Computational Linguistic Techniques for Sentence-Level Text Processing

Submitted by

Khaled Abdalgader M. Omar
BSc, MSc

A thesis submitted in total fulfilment
of the requirements for the degree of
Doctor of Philosophy

School of Engineering and Mathematical Sciences
Faculty of Science, Technology and Engineering

La Trobe University
Bundoora, Victoria, Australia

December 2011

Contents

List of Tables

List of Figures

Abstract

The availability of huge text collections stored in online repositories has created the potential of a vast amount of valuable information buried in those texts. This in turn has created the need for automated techniques of discovering new, relevant and useful information in those collections. The aim of this research is to develop computational linguistic techniques for sentence-level text processing. This is motivated by the belief that successfully being able to capture the interrelationships between sentence-level text fragments would lead to an increase in the breadth and scope of problems to which clustering, classification, and other text mining activities can successfully be applied.

The contributions of this thesis are four-fold. Firstly, a method is proposed for determining the relative importance of words in the sentences being compared. Secondly, a new similarity-based word sense disambiguation method is presented. The technique operates by computing the semantic similarity between WordNet glosses of the target word and the text fragment comprising all other words in the original sentence. This method is different from current methods, which compute the similarity only between words pairwise, and are thus limited, due to computational requirements, to using context from only a small window surrounding the target word. The third contribution is a new sentence similarity measure which incorporates word sense disambiguation and synonym expansion to provide a richer semantic context, thus enabling a more accurate estimate of semantic similarity between two sentences. The final contribution is a novel graph-based fuzzy relational clustering algorithm that can be applied to sentence level text clustering. The algorithm is based on expectation maximization; however, unlike conventional methods which represent clusters using Gaussian components, the proposed algorithm is based on estimating the likelihood of an object using the well-known PageRank algorithm. All of the proposed techniques perform favorably against the state-of-the-art related approaches, as evaluated on several benchmark datasets through different models of evaluation.

Declaration

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis submitted for the award of any other degree or diploma.

No other person's work has been used without due acknowledgement in the main text of the thesis.

The thesis has not been submitted for the award of any degree of diploma in any other tertiary institution.

I declare that the research in this thesis is my own original work during my PhD candidature under the supervision of Dr. Andrew Skabar, except where otherwise acknowledged in the text.

Khaled Abdalgader M. Omar

Khaled

December, 2011

Acknowledgments

All praise is due to Allah for granting me the strength and knowledge to complete this thesis.

It has been my very great privilege to know and work with my advisor Dr. Andrew Skabar in various capacities over the past three years. I would like to thank Dr. Skabar for his great guidance and timely support. I am greatly thankful that he would take me as a student, teach me the research methodology, guide me in choosing interesting and influential research topics, and encourage me. I feel very lucky to have had Dr. Skabar as an advisor, and look forward to having more opportunities to learn from and work with him in the future.

A special thanks to anonymous reviewers for giving feedback on my publications to help me become more confident in continuing the work.

Many thanks to my parents who are patiently waiting for me to come home to support them for the rest of their life. I am also grateful to my wife and family for their support, understanding and constant encouragements.

Last but not the least, I am greatly indebted to my friends and my colleagues at La Trobe University for their support. This research is supported by the Libyan government scholarship through its Embassy in Canberra. Many thanks go to them.

Publications

Portions of the material in this thesis have previously appeared in the following publications:

Abdalgader, K. and Skabar, A. 2010. Short-text similarity measurement using word sense disambiguation and synonym expansion. In *Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence*. (AI2010, Adelaide). Advances in Artificial Intelligence. 6464, 435-444.

Skabar, A. and Abdalgader, K. 2010. Improving sentence similarity measurement by incorporating sentential word importance. In *Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence*. (AI2010, Adelaide). Advances in Artificial Intelligence. 6464, 466-475.

Abdalgader, K. and Skabar, A. 2011. Unsupervised similarity-based word sense disambiguation using context vectors and sentential word importance. *ACM Transactions on Speech and Language Processing*.

Skabar, A. and Abdalgader, K. 2011. Clustering sentence-level text using a novel fuzzy relational clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*. IEEE computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.205>

Chapter 1

Introduction

The availability of large text collections stored in online repositories has created the potential of a vast amount of valuable information buried in those texts. This in turn has created the need for automated techniques for deriving relevant and useful information. Text mining can broadly be described as the process of deriving high quality information from text, where 'high quality' refers to some combination of relevance, novelty, and interestingness. In other words, the goals of text mining are to discover new information from some text [Hotho *et al*., 2005].

Tasks typically performed in text mining applications include text categorisation (i.e., classifying a text fragment as belonging to one or more predefined classes or categories) [Namburu *et al*., 2005]; text clustering (i.e., grouping text fragments according to their degree of similarity to one another) [Hatzivassiloglou *et al*., 2001; Radev *et al*., 2004], and text summarisation (i.e., producing a document summary which captures the main body of relevant content in some document or documents) [Chen *et al*., 2008; Kyoomarsi *et al*., 2008]. These tasks are not independent, and an activity focused on text summarisation, for example, may involve sub-tasks involving classification or clustering [Vidhya and Aghila, 2010].

Consider, for example, the problem of document summarisation [Zha, 2002; Aliguyev, 2009]. One approach to document summarisation is to identify the main themes or topics which characterise a document, and to then construct a summary of the document by appending, in a coherent manner, a description of each of those themes. Presumably, fragments of text that are similar to each other are more likely to relate to the same theme than fragments that are less similar. Thus, clustering, using both an appropriate similarity measure and an appropriate level of text fragmentation should provide a useful tool in allowing us to identify those themes.

An important question, then, is what unit of text should be used for clustering on tasks such as text summarisation: words, phrases, sentences, paragraphs, etc.? If the unit of fragmentation is too small (e.g., individual words), we may succeed in finding clusters of related words, but it will be difficult to recombine these words to create a summary. On the other hand, if the unit is too large (e.g., a paragraph or document), then we may not be able to clearly identify themes, since a paragraph may span a number of topics. Sentences are probably at about the right level of fragmentation since they tend to contain information about specific events, and are therefore more likely to provide a suitable context for identifying themes [Pederson, 2008; Naughton *et al*., 2006].

A second important question concerns representation; i.e., how should sentence-level text be represented in order that an appropriate similarity measure can be defined? Representations such as the Vector Space Model (VSM) [Salton, 1989], which are based solely on word co-occurrence and commonly used at the document level, are clearly not suitable at the sentence level, since two sentences may be about a similar topic, yet contain no words in common. For example, consider the sentences "The world is in economic crisis" and "The current dismal fiscal situation is global". Clearly these sentences have similar meaning, yet the only words they have in common are the stopwords 'is' and 'the', which contain little or no semantic information. At the sentence level term co-occurrence may be rare or even absent, and arises because the flexibility of natural language enables humans to express similar meaning using sentences that may be quite different not only in their structure, but also in regard to their component words [Bates, 1986]. Thus, at the sentence-level, we require a representation which is better able to capture the *semantic* content of sentences, thereby enabling a more appropriate similarity measure to be defined.

Various measures for sentence-level text similarity have been recently proposed [Li *et al.*, 2006; Mihalcea *et al.*, 2006; Ramage *et al.*, 2009; Achananuparp *et al.*, 2009]. All of these define the similarity of two sentences as being some function of the semantic similarities between their constituent words, and in this sense can be thought of as representing the sentences in a reduced vector space consisting only of the words in the two sentences being compared. The word's semantic similarities are typically based on word-to-word similarity measures derived either from distributional information from some corpora (corpus-based measures), or semantic information represented in external sources such as WordNet [Fellbaum, 1998] (knowledge-based measures). However, many words have more than one meaning (polysemy), and in order to accurately calculate the similarity between two sentences, it is therefore important to correctly identify the sense in which the constituent words are being used in those sentences. This problem is known as Word Sense Disambiguation (WSD). While WSD has a long history in the fields of computational linguistics and natural language understanding, there has been very little research reporting the incorporation of WSD into sentence similarity measurement [Ho *et al.*, 2010]. Moreover, the correct sense of a word needs to be determined in the context of the sentence in which it appears, and this presents yet another difficulty, since sentences provide only a very limited context.

The use of a reduced vector space for measuring sentence similarity also has important implications for sentence clustering. In particular, the fact that sentences are not represented in a common metric space means that existing prototype-based clustering algorithms such as *k*-Means [MacQueen, 1967], Isodata [Ball and Hall, 1967] and Fuzzy *c*-Means [Dunn, 1973; Bezdek, 1981], which accept *attribute data* as input (i.e., rectangular data where rows represent the objects to be clustered and columns represent the attributes of those objects; e.g., Term Frequency-Inverse Document Frequency (TF-IDF) scores in the case of document clustering), are generally not applicable. Rather, sentence clustering must be based on *relational data*; i.e., input data in the form of a square matrix $S = \{s_{ij}\}$, where s_{ij} is the (pairwise) relationship between the *i*th and *j*th data object (i.e., sentence). Sentence clustering is further complicated by the fact that most documents will contain interrelated topics or themes, and many sentences will be related to some degree to a number of these. This means that ideally a clustering algorithm should be able to identify soft, or *fuzzy*, clusters. The topic of interest therefore is *fuzzy relational clustering*; i.e., fuzzy clustering based on (pairwise) relational input data is required.

The research described in this thesis is motivated by the belief that successfully being able to capture such fuzzy relationships will lead to an increase in the breadth and scope of problems to which clustering, classification, and other text mining activities can successfully be applied at the sentence level. However, the performance of any clustering or classification algorithm will be limited by the quality of the input data, and in the case of sentence-level text, performance will depend fundamentally on the quality of the sentence similarity measure that is used. Thus, while the thesis makes contributions in a number of areas, the issue of sentence similarity holds a central position. Thus the main question that the thesis addresses is:

1. *Can sentence similarity measurement be improved through incorporating WSD and context expansion*?

This, in turn, relies on an appropriate WSD technique. Since sentences provide relatively little context, it is therefore important to make optimal use of the context that is available. This leads to the following question:

2. *Can WSD performance be improved by better utilising the context provided by the surrounding words?*

Sentence similarity measures need to be carefully evaluated. While evaluation is often performed *in vitro* (stand-alone) using standard datasets designed specifically for this purpose, it is important that the techniques also be evaluated *in vivo*, i.e., on end-to-end tasks such as clustering. While a number of clustering algorithms are available, most existing relational clustering algorithms are not capable of identifying fuzzy clusters. Therefore,

3. *Can a relational clustering algorithm be devised that is better able to capture the complex and subtle interrelationships between text objects at the sentence level?*

Figure 1.1 shows the relationship between these questions.

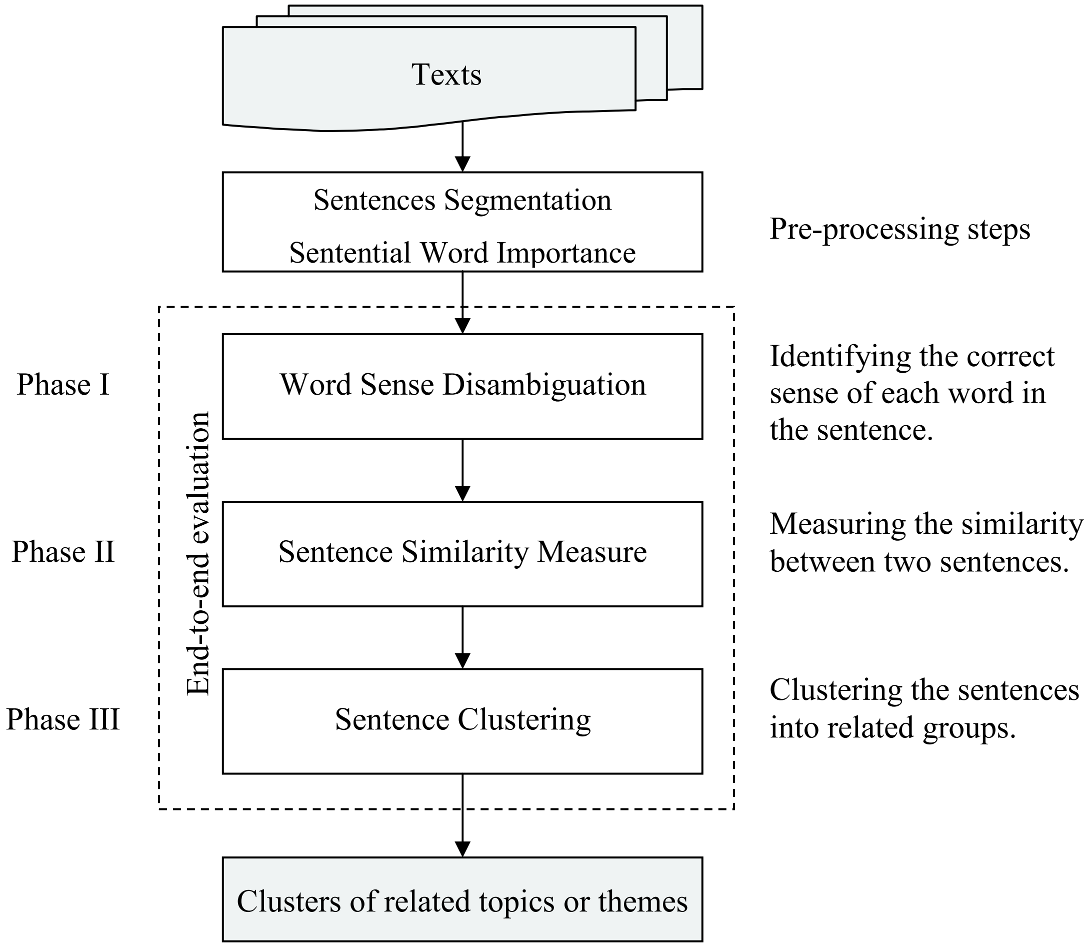
Texts
Sentences Segmentation
Sentential Word Importance
Pre-processing steps
Phase I
Word Sense Disambiguation
Identifying the correct sense of each word in the sentence.
Phase II
Sentence Similarity Measure
Measuring the similarity between two sentences.
Phase III
Sentence Clustering
Clustering the sentences into related groups.
End-to-end evaluation
Clusters of related topics or themes

Figure 1.1: Diagrammatic view of the relationship between research questions.

1.1 Thesis Contributions

The thesis has four main contributions.

1. **A method for determining Sentential Word Importance (SWI), and incorporating it into sentence similarity measurement**

Existing sentence similarity measures such as those of Li *et al*. (2006) and Mihalcea *et al*. (2006) typically incorporate an IDF (inverse document frequency) scaling whereby words that occur less frequently in some large corpus are weighted more strongly in the sentence similarity calculation than words that occur more frequently. This thesis shows how graph centrality measures can be used to assign a numerical measure of importance to each word of a sentence, based only on its relationship to the other words in the sentence (without reference to any external corpus), and how these values can be incorporated within various existing sentence similarity measures. The method has previously been reported in Skabar and Abdalgader (2010).

2. **A novel similarity-based Word Sense Disambiguation (WSD) method**

Current WSD methods such as those described in Kilgarriff and Rosenzweig (2000), Patwardhan *et al*. (2003), Sinha and Mihalcea (2007) and Navigli and Lapata (2010) are based on measuring pairwise similarity between *words*. This thesis presents a new similarity-based WSD method that determines the sense of a target word by measuring the semantic similarity between its WordNet glosses (i.e., dictionary definitions) and the context provided by all remaining words in the given text fragment, which is referred to as the ‘context vector’. The correct sense of the target word is identified as the sense for which the semantic similarity between gloss vector and context vector is highest. This enables it to utilise a higher degree of semantic information than current approaches, and is more consistent with the way that human beings disambiguate; that is, by considering the greater context in which the word appears. The thesis also shows how performance can be further improved by incorporating a preliminary step in which the relative importance of words within the original text fragment is estimated, thereby providing an ordering that can be used to determine the sequence in which words should be disambiguated. This contribution has previously been reported in Abdalgader and Skabar (2011).

3. A novel sentence similarity measure

Sentence similarity measures such as those proposed by Li *et al*. (2006) and Mihalcea *et al*. (2006) do not fully utilise the semantic information available from lexical resources such as WordNet. The third contribution of this thesis is a new sentence similarity measure that utilises word sense disambiguation, in conjunction with synonym expansion, to create an enriched semantic context, thus enabling a more accurate estimate of semantic similarity between two sentences. *In vivo* evaluation on a number of clustering tasks using a variety of clustering algorithms demonstrates that the measure leads to significantly better clustering performance than baseline measures that do not incorporate WSD and synonym expansion. This similarity measure has previously been reported in Abdalgader and Skabar (2010).

4. A novel fuzzy relational clustering algorithm

The final contribution of the thesis is a novel fuzzy relational clustering algorithm. Inspired by the mixture model approach, the data is modelled as a combination of components. However, unlike conventional mixture models, which operate in a Euclidean space and use a likelihood function parameterised by the means and covariances of Gaussian components, the algorithm abandons use of any explicit density model (e.g., Gaussian) for representing clusters. Instead, a graph representation is used, in which nodes represent objects, and weighted edges represent the similarity between objects (sentences). Cluster membership values for each node represent the degree to which the object represented by that node belongs to each of the respective clusters, and mixing coefficients represent the probability of an object having been generated from that component. By applying the PageRank [Brin and Page, 1998] graph centrality algorithm to each cluster, and interpreting the PageRank score of an object within some cluster as a likelihood, the Expectation-Maximisation (EM) framework [Dempster *et al.*, 1977] is then used to determine the model parameters (i.e., cluster membership values and mixing coefficients). The result is a fuzzy relational clustering algorithm which is generic in nature, and can be applied to any domain in which the relationship between objects is expressed in terms of pairwise similarities. The fuzzy clustering algorithm has previously been reported in Skabar and Abdalgader (2011).

1.2 Thesis Outline

The thesis is organised as follows:

Chapter 2 provides relevant background on topics in linguistic computing, focussing particularly on the areas of word sense disambiguation, text similarity measurement and text clustering. It also describes the standard benchmark datasets used in the thesis, and contains a section on WordNet.

Chapter 3 demonstrates how graph centrality methods can be used to determine sentential word importance, and how this can be incorporated into existing sentence similarity measures.

Chapter 4 presents a new similarity-based WSD algorithm based on the notion of context vectors, and demonstrates how its performance can be improved through using sentential word importance to determine word disambiguation order.

Chapter 5 presents a new sentence similarity measure which uses WSD and synonym expansion to provide a richer semantic context to measure sentence similarity.

Chapter 6 presents a new fuzzy relational clustering algorithm based on eigenvector graph centrality. Performance of the algorithm is compared against several other clustering algorithms on a range of sentence clustering tasks, using a variety of sentence similarity measures. Thus the chapter also serves the purpose of evaluating the proposed similarity measure on *in vivo* task.

Chapter 7 concludes the thesis with a review and discussion of the contributions of the research, and a discussion of future research directions.

Appendix A presents the excerpts of experimentally used benchmark datasets for WSD and sentence similarity measurement experiments.

Appendix B shows the compiled datasets for the clustering experiments that have been presented in Chapter 6.

Chapter 2

Background and Related Work

This chapter presents background into computational linguistic techniques relevant to text mining. The chapter starts with a brief introduction to the field of linguistic computing. Section 2.2 discusses the common approaches to word sense disambiguation (WSD). Section 2.3 reviews sentence-level text similarity approaches. Section 2.4 presents the clustering algorithms that can be used to cluster sentences. Section 2.5 details the experimentally used benchmark datasets. Section 2.6 describes WordNet and Section 2.7 draws some conclusions.

2.1 Topics in Linguistic Computing

The main objective of computational linguistics is to enable computers to be used as aids in analysing the properties of linguistic theories, and to understand—by analogy with computers—more about how humans process natural language [Wiebe, 1994; Bolshakov and Gelbukh, 2004]. By understanding language processes in procedural terms, computers can gain the ability to generate and interpret natural language [Allen, 1995]. This would make it possible for computers to perform highly useful linguistic tasks including part-of-speech (POS) tagging and spell checking, text similarity measurement, text clustering, automated question answering, etc. These tasks are very complex and involve processing on many levels (morphological, semantic, etc.). Recently, the area of computational linguistics has developed dramatically for several reasons: the growth of online textual context (leading to rich sources of information); the rise in computer capability (text processing is computationally challenging), and the development of linguistic resources in digital format [Jurafsky and Martin, 2009].

2.1.1 Natural Language Processing (NLP)

Natural language processing is concerned with the development of computational techniques for analysing, understanding and representing natural human language for the purpose of achieving human-like computer systems. It was founded in the early 1960s as a sub-field of artificial intelligence and linguistics, with the aim of studying problems in the automatic generation and understanding of natural language [Charniak and Eugene, 1984]. During its early decade, research in NLP was based on symbolic models including logic and formal rules; however, in recent years there has been a shift in interest towards *statistical* and *linguistic* approaches relying on analysis of large amounts of textual and lexical resources with an acceptable level of efficacy [Baeza-Yates, 2004].

Statistical approaches (also known as *empirical approaches*) apply several mathematical theories and often employ a large collection of texts (i.e., corpora) to develop human-like language processing methods for a range of applications. These include speech recognition [Jelinek, 1999], machine translation [Brown *et al*., 1990; Brown and Frederking, 1995], information retrieval (IR) [Ponte and Croft, 1998; Berger and Lafferty, 1999] and many more [Rabiner and Juang, 1986; Charniak, 1995; Rosenfield, 2000; Mihalcea and Moldovan, 1999]. The statistical approaches are typically not based on linguistic notions, but rather, on exploring the actual instances of linguistic phenomena provided by text corpora. Crucial tasks using the statistical approaches are part-of-speech tagging and alignment.

Linguistic approaches, in contrast, perform deep analysis of linguistic phenomena [Liddy, 1998; Feldman, 1999], and are inspired by the belief that computers can be made to be human-like through providing basic linguistic knowledge and reasoning mechanisms, explicitly encoded in rules or other forms of representation. Linguistic approaches have been used in recent years in a variety of NLP applications. These include text classification [Scott and Matwin, 1998; Gee and Cook, 2005], word sense disambiguation [Montoyo *et al*., 2005], lexical acquisition [Hearst, 1992] and many more [Liddy, 2010; Kazakov *et al*., 1999]. In contrast to statistical approaches, linguistic approaches do not use corpora as the main source of evidence.

Both linguistic and statistical approaches exhibit different characteristics, therefore some tasks may be better tackled with one approach, while other tasks by another [Liddy, 2010]. In some cases, for some particular tasks, one approach may prove adequate, while in other cases the tasks can be so complex that it might not be possible to select a single best approach. As a result, hybrid techniques that utilise the strengths of each approach also exist, and attempt to address NLP more effectively and in a more flexible manner [Pazienza *et al*., 2005].

2.1.2 Levels of Linguistic Processing

Several levels of linguistic processing can be identified: *morphological*, a componential analysis of words, including prefixes, suffixes and roots; *lexical*, a word-level analysis that includes lexical meaning and part-of-speech analysis; *syntactic*, an analysis of words in a sentence in order to uncover the grammatical structure; and *semantic*, which is concerned with determining the possible meanings of words and how these combine to form the meaning of a sentence.

The morphological processing level (also known as *stemming*) is applied in many NLP applications to reduce different variants of the same word with different endings (i.e., affix) to a common root form [Paice, 1996]. The root is a primary lexical form of a word, and carries the most significant aspects of semantic content it cannot be reduced into smaller constituents. For example, the word ‘stem’ is the root or base form of the words ‘stemmer’, ‘stemming’ and ‘stemmed’. Various stemming algorithms have been proposed in recent years [Porter, 1980; Krovetz, 1993]. The most widely used algorithm—possibly due to its efficiency—is the Porter stemmer [Porter, 1980], which has become a standard in text retrieval systems, and is the stemmer used in all experiments conducted in this thesis. Operations such as *stopword* removal (i.e., removal of words such ‘a’ and ‘the’, which carry little semantic meaning) might also be carried out as part of the morphological processing task. Most stopwords lists[1] for English language developed in recent years are usually based on frequency statistics of a large corpus [Van Rijisbergen, 1975; Francis and Kucera, 1982; Fox, 1990].

The lexical level of linguistic processing is concerned with interpreting the meaning of individual words. The most essential types of processing that contribute to word-unit understanding are identifying the part-of-speech tag, and identifying the sense of a polysemous word based on the context in which it occurs. In part-of-speech processing, words are assigned a tag that represents their part-of-speech function in the given context. Parts of speech include nouns, verbs, adjectives, adverbs, pronouns, conjunction and their sub-categories. Note that many words can have more than one part-of-speech associated with them. For example, ‘bank’ can be a noun or verb, depending on its context. Word sense disambiguation (WSD) is the process of identifying which sense (meaning) of a word is intended in some given context. For example, consider the distinct senses that exist for the word ‘bass’: one as a type of fish, and the other as a tone of music. Section 2.2 reviews the problem of word sense disambiguation.

The lexical level of language is evidenced in the knowledge contained in lexical resources such as thesauri. A lexical resource may be very simple, containing only the

[1] Usual candidate words of this list are articles, prepositions, and conjunctions, although specific nouns, verbs or other grammatical types could also be included if they are considered to be of low importance in the specific domain.

words and their parts of speech, or it may be more complex, containing information on the semantic class of the words. Depending on the type of languages that are addressed, the lexicon may be qualified as monolingual, bilingual or multilingual. It is possible also to build and manage a lexical resource consisting of different lexicons of the same language; for example, one dictionary for general words, and one or more dictionaries for different specialised domains. Section 2.6 presents a review of the lexical resources that have been used in this thesis.

The syntactic level is focused on analysing the part-of-speech tagging produced from the lexical level, and can assign phrase and clause brackets. The result of this analysis is a structural representation of the processed sentence that reveals the syntax relationships between the words. There are various grammars that can be analysed, and which will, in turn, impact on the selection of a parser. Most NLP applications require a full parse of the input text to achieve adequate performance. In text retrieval systems, for example, syntactically identified phrases extracted from the query can provide better searching keys for matching against similarly bracketed documents [Liddy, 1998; Feldman, 1999].

The semantic level of linguistic processing utilises the possible meanings (senses) of the words within some context (e.g., sentence). This level of processing can include the semantic disambiguation of polysemous words, context expansion by addition of all available synonyms corresponding to the context words, etc. Note that while WSD can be performed at the syntactic level, *meaning* of the word can only be ascertained at the semantic level. NLP has to use both syntactic and semantic levels to determine the meaning of words from the context of the sentences.

2.2 Word Sense Disambiguation (WSD)

This section reviews the problem of WSD, existing approaches, knowledge resources that assist the disambiguation process, and performance evaluation criteria.

Many words have more than one possible sense, depending on the context in which they appear. For example, if the words ‘deposit’, ‘money’ and ‘loan’ appear near the word ‘bank’, humans can easily identify that the intended sense of ‘bank’ is the financial institution, and not the slope beside a body of water. Word sense disambiguation—the process of identifying the appropriate senses (meanings) of words as they occur in some text fragment (e.g., a sentence)—is an intermediate and fundamental task in many natural language processing applications. These include information retrieval, where we are supplied a query consisting of a few words, and the objective is to retrieve documents pertinent to the query from a collection of documents [Stokoe, 2005]; question answering, in which the system attempts to determine the coherent meaning of a question, and then collates information to provide a correctly formed answer to the intended question [Ramakrishnan *et al*., 2003]; machine translation, where the system automatically translates a collection of words into the target language [Vickrey *et al*., 2005; Carpuat and Wu, 2007; Chan *et al*., 2007]; and text mining and text summarisation, which require broad-coverage language understanding [Barzilay and Elhadad, 1997].

Naturally, it is impossible to disambiguate a word in isolation. This is because there is no context to distinguish between its senses. Thus one or more surrounding words are necessary to provide enough evidence to identify the correct meaning of a word. This is known as *local context* or *sentential context*.

The text fragment can be viewed as a sequence of words $W = \{w_i \mid i{=}1..N\}$, where N is the number of target words in W. WSD, then, is defined as the process of labelling the appropriate senses to all words in W, associating a linking A from words to senses, such that $A(i) \subseteq t_{w_i}$, where t_{w_i} is the set of senses encoded in a sense inventory for w_i, and $A(i)$ is that subset of the senses of w_i which are appropriate in W. The A can label more than one sense to each word $w_i \in W$, although typically only one sense is assigned; i.e., $|\, A(i)| = 1$. In this thesis we will assume that only one label is to be assigned to each polysemous word.

WSD can be considered as a classification task in which word senses are the classes, and a classifier is used to assign each word to one class, depending on the evidence from its context and sense inventory. However, there are two variant settings of WSD: lexical sample, and all-words. In the lexical sample setting, the WSD algorithm is required to disambiguate a limited set of words, usually words in the same part-of-speech category. In contrast, in the all-words setting, the WSD algorithm must disambiguate all words appearing in the given text fragment, irrespective of their part-of-speech categories. The next section discusses existing approaches for WSD.

2.2.1 Word Sense Disambiguation Approaches

Various WSD methods have been proposed in recent years [Lesk, 1986; Resnik, 1995; Banerjee and Pedersen, 2003; Patwardhan *et al*., 2003; Snyder and Palmer, 2004; Navigli and Velardi, 2005; Mihalcea, 2005; Pradhan *et al*., 2007; Sinha and Mihalcea, 2007; Navigli, 2008; Navigli and Lapata, 2010; Agirre and Soroa, 2009], and can be broadly categorized as belonging to one of two families: *corpus-based* methods, and *knowledge-based* methods.

Corpus-based methods utilise *supervised* learning techniques to induce a classifier from a corpus of training data consisting of a set of labeled words, in which the label indicates the sense in which the word is being used. Once a classifier has been created by extracting the syntactic and semantic features, it can then be used to predict the sense of the target word in novel sentences. In contrast, knowledge-based methods are usually *unsupervised* and do not require any such corpora, relying instead on external lexical resources such as dictionaries or thesauri [Navigli, 2009]. While corpus-based methods have generally been found to perform more accurately than knowledge-based methods [Snyder and Palmer, 2004; Pradhan *et al*., 2007], the fact that a separate classifier must be induced for every word severely limits the coverage of these methods. This is because corpus-based methods usually require large amounts of hand-labeled data in order to obtain reliable results [Yarowsky and Florian, 2002], and labeling such data is an intensive and time consuming process. Therefore, despite their lower overall accuracy, knowledge-based methods tend to be preferred on account of the broad coverage that they achieve. This thesis focuses on knowledge-based WSD methods only.

Knowledge-based WSD methods fall into two main groups: *similarity-based* methods, and *graph-based* methods. Similarity-based methods determine the sense of a polysemous word by computing the similarity between each of its possible senses and the words in the surrounding context. The correct sense of the target word is then assumed to be that for which the similarity is greatest. Graph-based methods, however, usually build a semantic structure (i.e., a graph) representing all available senses of all of the words being disambiguated. The nodes in this graph correspond to these senses and the edges represent the lexical relation (e.g., synonymy, antonymy, hyperonymy, etc.) between them. Graph centrality methods are then typically used to determine which nodes are more important (i.e., central) within the graph, and these are considered to be the correct senses of the target words. Because they disambiguate all words in a text fragment simultaneously by exploiting semantic similarities across word senses, graph-based methods usually achieve higher performance than similarity-based methods, which disambiguate words individually, usually without considering the senses assigned to surrounding words [Navigli and Lapata, 2010]. However, the main disadvantage of graphical methods is their high computational complexity.

Similarity-based Approaches

Similarity-based methods are inspired by two linguistic distributional hypotheses [Harris, 1954]. The first is that words that are similar in meaning tend to appear nearby in the same text fragment. The second is that actual senses can be identified by finding shared words (i.e., word overlap) in their dictionary definitions. The first word sense disambiguation algorithm that capitalised on these hypotheses is due to Lesk

(1986). This method determines the sense of a target polysemous word by calculating the word overlap between the glosses (i.e., dictionary definitions) of two or more target words. The correct senses of the target words are assumed to be those whose glosses have the greatest word overlap. Formally, in the case of two words w_i and w_j, Lesk score is defined as:

$$Score_{Lesk}(S_i^m, S_j^n) = |\, gloss(S_i^m) \cap gloss(S_j^n)\, | \tag{2.1}$$

where *gloss*(S_i^m) is the bag of content words in the gloss of sense *m* of a word w_i, and *gloss* (S_j^n) is the bag of content words in the gloss of sense *n* of a word w_j. The senses which score the highest value from the above calculation are assigned to the respective words. For example consider the task of disambiguating the words ‘software’ and ‘virus’, each with possible associated glosses. The first gloss for ‘software’ and the third gloss for ‘virus’, (according to WordNet [Fellbaum, 1998]) have the largest overlap among all available gloss combinations, with two words in common: ‘computer’ and ‘program’. Therefore, these are the senses selected by Lesk’s method.

While the Lesk (1986) method is feasible when the context is small (e.g., two words) it leads to combinatorial explosion as the number of words increases. In a two-word context, the number of gloss overlap calculations is $|senses(w_i)| \cdot |senses(w_j)|$, where $senses(w_i)$ denotes the set of possible senses for word w_i. In an *n*-word context this increases exponentially to $|senses(w_i)| \cdot |senses(w_j)| \cdot \ldots \cdot |senses(w_n)|$. Consider for example, the sentence “All fish in the river of the south bank have been infected by the virus”, with six open class words, each with multiple possible WordNet senses: fish(6), river(1), south(7), bank(18), infected(5), virus(3). A total of 11,340 sense combinations are possible for disambiguating these words. For this reason, a simplified version [Cowie *et al*., 1992; Kilgarriff and Rosenzweig, 2000] of this approach is commonly used, in which the sense for word w_i is selected as the one whose gloss has the greatest overlap with the words in the context of w_i. That is,

$$Score_{LeskVar}(S_i^m) = | gloss(S_i^m) \cap context(w_i) | \qquad (2.2)$$

where *context* (w_i) is the bag of words in a context (e.g., sentence) that contains word w_i. Using the simplified approach, the number of overlap calculations in the above example is reduced from 11,340 to 40. Note, however, that whereas the original Lesk method disambiguates all words simultaneously, the simplified approach disambiguates each word individually, and this would normally be expected to lead to inferior performance to the original Lesk method.

Lesk's algorithm suffers from the fact that dictionary glosses are often quite brief, and may not include sufficient vocabulary to identify appropriate senses. To alleviate this problem, Banerjee and Pedersen (2003) proposed an extended gloss overlap method based on the use of WordNet. Rather than only considering the glosses of the words in the original context, the concept hierarchy of WordNet is exploited to expand those glosses to include glosses of those words to which they are related through lexical relations (e.g., hyperonymy, meronymy, etc.). This method starts by selecting a context window that contains the target word and small number of content words that are known to WordNet. The correct sense for the target word w_i is then selected as the one whose expanded gloss has the greatest overlap with the words in the context window around w_i. Note that the reason for limiting the context to a small window is to reduce the computational requirements.

While Lesk's algorithm and many of its variants are based on the notion of gloss overlap, Patwardhan *et al*. (2003), following Rada *et al*. (1989) and Resnik (1995), take the view that gloss overlap is just one of many possible measures of semantic similarity, and propose replacing the use of gloss overlap with a semantic word-to-word measure based on the WordNet concept hierarchy (i.e., similarity between word-sense pairs). Their method is inspired by the natural property of human language that words in a text must be related in meaning for the text to be coherent [Halliday and Hasan, 1976]. This means that words that have shortest semantic distance are usually closely related in meaning. Patwardhan *et al*. (2003) experiment with five word-to-word semantic similarity measures defined on WordNet (these measures are described in Section 2.3.3) and find that the best performance is obtained using the J&C [Jiang and Conrath, 1997] measure. Note that for computational reasons, they too limit the context to a small window around the target word. The differences between Lesk's method and its variants are illustrated in Figure 2.1.

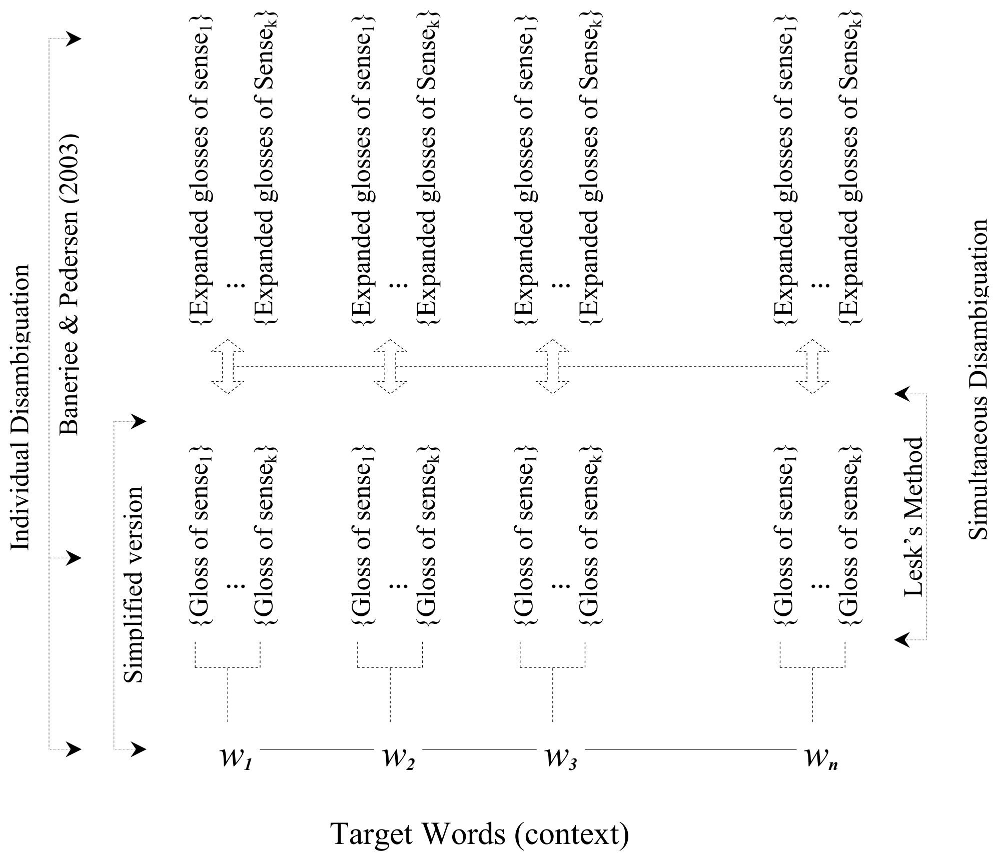
Individual Disambiguation
Banerjee & Pedersen (2003)
Simplified version
{Expanded glosses of sense$_1$}
...
{Expanded glosses of Sense$_k$}
{Gloss of sense$_1$}
...
{Gloss of sense$_k$}
Lesk's Method
Simultaneous Disambiguation
w_1
w_2
w_3
w_n
Target Words (context)

Figure 2.1: Differences between Lesk's method and its variants.

The method proposed by Agirre and Rigau (1996) is also based on the notion of word-word semantic similarity, but in this case, rather than measuring similarity directly, similarity is measured by considering the conceptual density of word senses, with the correct sense of a target word identified as that which is in the area of highest density of the words in the context. Density is measured in terms of hypernymy relations, which means that more specific areas of the hierarchy (i.e., senses deeper in the WordNet hierarchy) are considered closer than more general areas (i.e., senses at a shallower level). Once again, computational requirements necessitate that words are disambiguated individually; i.e., without considering the sense assigned to the words in the context.

Graph-based Approaches

Graph-based methods have attracted much recent attention, mainly because they have narrowed the performance gap between supervised and unsupervised methods [Navigli and Velardi, 2005; Mihalcea, 2005; Sinha and Mihalcea, 2007; Agirre and Soroa, 2009; Navigli and Lapata, 2010]. These methods operate by constructing a graph representing all available senses of the words being disambiguated, with nodes in the graph corresponding to senses, and edges representing the lexical relation (e.g., synonymy, antonymy, hyperonymy, similarity, etc.) between them. The graph structure is then exploited to determine the importance of each node.

As an example, consider the approach due to Sinha and Mihalcea (2007). Given a set of N words, $W = \{w_i \mid i = 1..N\}$, and their corresponding senses $S_{w_i} = \left\{S_{w_i}^k \mid k = 1...a_{w_i}\right\}$, where a_{w_i} is the number of senses for w_i, a graph $G = (V, E)$ is defined such that there is a node $v \in V$ for every available sense $S_{w_i}^k$, $i = 1...N$, $k = 1..a_{w_i}$. Edges $e \in E$ map the dependency (using one of six known word-to-word similarity measures) between pairs of senses, forming a graph over the text fragment. One of four graph centrality algorithms (e.g., indegree, closeness, betweenness [Freeman, 1979], and PageRank [Brin and Page, 1998]) is then used to assign to each node a score reflecting its importance within the graph. The correct sense for a target word is then identified as the sense corresponding to the node with the highest score of all senses of that target word. The general representation of the graph methods is depicted in Figure 2.2.

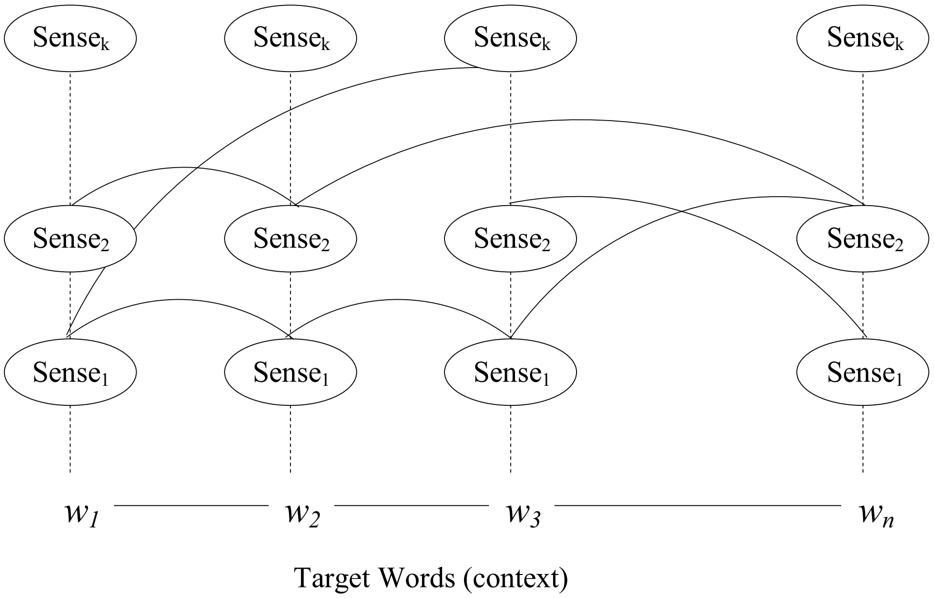

Figure 2.2: General representation of disambiguation process for graph-based methods.

The graph structure in Sinha and Mihalcea's (2007) method may not be fully connected as not all word sense pairs may be semantically or syntactically related, and this low connectivity may lead to disambiguating words without taking into account neighbouring related words, thereby possibly resulting in an incoherent set of meanings (i.e., a loss in coverage). To overcome this problem, Navigli and Lapata (2010) recently proposed a method based on the use of depth-first search to explore the lexical relations connecting word senses. Using back-tracking, search proceeds until all the possible relations between nodes has been discovered. Importantly, the graph in this case is built directly from WordNet lexical relations, thus allowing a greater relational connectivity between words than is the case for the original approach of Sinha and Mihalcea (2007), which only considers current word senses. Various other graph-based approaches have also been proposed, and include Tsatsaronis *et al.* (2010) and Agirre and Soroa (2009), the latter of which extended the graph using eXtended WordNet [Mihalcea and Moldovan, 2001].

While graph-based methods generally perform better than similarity-based approaches, their main disadvantage is their computational expense [Navigli and Lapata, 2010; Tsatsaronis *et al.*, 2010]. To demonstrate, let *a* be the average number of senses (i.e., nodes) per word, *l* the maximum semantic path length between any two nodes, and *N* the number of words to be disambiguated. The number of combinations required to construct the graph is at least $O(N.d^{l+1})$, and note that the value of *l* can differ significantly from one method to another. But in addition to the cost of constructing the graph, there is also the cost of applying one of the graph centrality algorithms. For example, the time complexity of PageRank is $O(N^2.a^{3/2*l+3})$ in the worst case [Tsatsaronis *et al.*, 2010]. Similarity-based methods generally involve far fewer operations. For example, the time complexity of Patwardhan *et al.*'s (2003) approach is only $O(a^{N_w})$, where N_w is the number of words in the window around the target word.

2.2.2 Knowledge Resources for Word Sense Disambiguation

In order to perform WSD, knowledge resources (i.e., lexical resources) are required. This is because without knowledge, it would be impossible for either humans or computers to identify the correct sense of a word. These resources provide lexical knowledge which is essential to assigning appropriate senses to words, and can be broadly categorised as belonging to one of two families: structured resources (e.g., dictionaries, thesauri and ontologies), and unstructured resources (e.g., labeled corpora and unlabeled corpora). A comprehensive review of WSD knowledge resources can be found in Ide and Veronis (1998), Litkowski (2005) and Agirre and Stevenson (2006).

WordNet can be viewed as a lexical ontology, which has been widely used in the field of WSD. In WordNet, word information is organised according to word senses. This is different to dictionaries, which organise words according to morphology. All English words are organised into synonym sets (*synsets*), each representing one underlying lexical concept. Different semantic relations connect the synsets, such as hypernymy, hyponymy, synonymy and antonymy. WordNet is particularly well-suited for WSD, since it is designed based on word sense information. Since WordNet is used as a lexical knowledge resource in all experiments reported in this thesis, a more detailed description of WordNet is provided in Section 2.6.

2.2.3 Evaluation of Word Sense Disambiguation Methods

WSD methods are usually evaluated using a stand-alone (i.e., *in vitro*) evaluation model [Kilgarriff and Palmer, 2000; Palmer *et al.*, 2006; Ides and Veronis, 1998], and several standard datasets have been constructed specifically for this purpose. For example, the Senseval/Semeval[1] campaign provides a shared task with a variety of datasets and sense inventories for all-words and lexical sample settings in different languages. The SemCor [Miller *et al.*, 1993], Senseval-2 [Palmer *et al.*, 2001] and Senseval-3 [Snyder and Palmer, 2004] datasets are the most common standard datasets, and are described further in Section 2.5.1. Detailed information on the Senseval competitions can be found in Martinez (2004) and Palmer *et al.* (2006).

One of the main motivations, however, for performing WSD is to improve the performance of real NLP applications. The evaluation of WSD as a task embedded in NLP applications is known as end-to-end (i.e., *in vivo*) evaluation [Jurafsky and Martin, 2009]. This model of evaluation is always a complicated issue due to the

[1] A series of international WSD competitions (http://www.senseval.org), organized by the ACL-SIGLEX that has been held every three years since 1998.

required incorporation of WSD into complete working systems, and for this reason WSD methods are usually evaluated as *in vitro*, independent of any particular application. To the best of author's knowledge, the research described in this thesis is the first *in vivo* evaluation of WSD within the task of sentence-level text similarity measurement.

2.3 Text Similarity Measurement

This section first briefly reviews the vector space model—the most common model for text representation. It then focuses on sentence-level text similarity measurement.

2.3.1 Vector Space Model

The most common text representation model is the vector space model [Salton, 1989], which has been widely used in traditional IR approaches [Salton and McGill, 1983; Baeza-Yates and Ribeiro-Neto, 1999; Manning *et al.*, 2008]. In the vector space model, each document is represented as a vector in an *n*-dimensional space, where *n* is the number of distinct words (i.e., terms) in the collection [Salton, 1971; Baeza-Yates and Ribeiro-Neto, 1999; Witten *et al.*, 1999; Manning *et al.*, 2008]. The similarity S($\mathbf{V}_1$, $\mathbf{V}_2$) between two document vectors $\mathbf{V}_1 = [w_{11}, w_{12}, w_{13}, \ldots, w_{1n}]$ and $\mathbf{V}_2 = [w_{21}, w_{22}, w_{23}, \ldots, w_{2n}]$ can be calculated using the dot product:

$$S(\mathbf{V}_1, \mathbf{V}_2) = \mathbf{V}_1 \bullet \mathbf{V}_2 = \sum_{t=1}^{n} w_{\mathrm{V}_1,t} \times w_{\mathrm{V}_2,t} \ , \qquad (2.3)$$

where $w_{\mathrm{V}_1,t}$ is the weight of word *t* in document 1 ($\mathbf{V}_1$), and $w_{\mathrm{V}_2,t}$ is the weight of word *t* in document 2 ($\mathbf{V}_2$). A word weight is a value which reflects the word's importance in the document.

Many different methods of computing weighs have been employed, the most common being TF-IDF (*Term Frequency—Inverse Document Frequency*) weighting. The *term frequency* (TF) of a word is its frequency in the document, and the *inverse document frequency* (IDF) is a measure of the general importance of the word over a large collection of documents, calculated by taking the logarithm of the total number of documents divided by the number of documents containing the word (term). Words that do not appear in a document receive a weight of 0.

To avoid bias towards longer documents, the dot product can be divided by the Euclidean length of the document vectors, which defines the cosine of the angle between the two document vectors. This measure is known as the *cosine* similarity measure [Salton and Lesk, 1968], and is defined as:

$$S(\mathbf{V}_1,\mathbf{V}_2)=\frac{\mathbf{V}_1 \bullet \mathbf{V}_2}{|\mathbf{V}_1||\mathbf{V}_2|}=\frac{\sum_{t=1}^{n} w_{\mathbf{V}_1,t}\times w_{\mathbf{V}_2,t}}{\sqrt{\sum_{t=1}^{n} w_{\mathbf{V}_1,t}\times\sum_{t=1}^{n} w_{\mathbf{V}_2,t}}}. \tag{2.4}$$

While the vector space model is very efficient because of its simplicity, the number of distinct words exceeds thousands even after many stopwords are removed (i.e., problem of data sparseness). To tackle this problem, various dimension reduction models have been proposed [Jeon *el al*., 2001; Landauer *el al*., 1998; Burgess, 1998; Turney, 2001; Kanerva *el al*., 2000; Steyvers *el al*., 2004; Lemaire and Denhiere, 2004]. Latent Semantic Indexing (LSI) [Landauer *el al*., 1998] is one of the most commonly used models, and is based on the assumption that words that are used in the same context tend to have similar meanings. A matrix table containing word counts, in which rows represent distinct words and columns represent each text fragment, is constructed from each document, and Singular Value Decomposition (SVD) [Wall *et al*., 2003] is then employed to reduce the number of columns while keeping the semantic similarity between rows. The semantic similarity is then calculated by taking the cosine between the two rows. If words from rows are similar, then values of 1 are assigned; otherwise similarity values are 0. The main disadvantage of LSI is that it is computationally expensive, since it depends heavily on SVD.

2.3.2 Sentence-Level Text Similarity Approaches

The vector space model has been successful in IR because it is able to adequately capture much of the semantic content of document-level text. This is because documents that are semantically related are likely to contain many words in common, and thus are found to be similar according to popular vector space measures, which are based on word co-occurrence [Manning *et al*., 2008]. However, while the assumption that (semantic) similarity can be measured in terms of word co-occurrence may be valid at the document level, the assumption does not hold for small-sized text fragments such as sentences, since two sentences may be semantically related despite having few, if any, words in common.

Measuring the similarity between sentence-level text fragments is a fundamental function in many textual applications. These include text mining and text summarisation, which usually operate at the sentence or sub-sentence level [Atkinson-Abutridy *et al*., 2004; Erkan and Radev, 2004]; question answering, where it is necessary to calculate the similarity between a question-answer pair [Bilotti *et al*., 2007; Mohler and Mihalcea, 2009]; and image retrieval, where we are interested in the similarity between a query and an image caption [Coelho *et al*., 2004].

Various linguistic measures for sentence similarity have been proposed in recent years [Li *et al*., 2006; Mihalcea *et al*., 2006; Zhao *et al*., 2006; Metzler *et al*., 2007; Islam and Inkpen, 2008; Feng *et al*., 2008; Ramage *et al*., 2009; Achananuparp *et al*., 2009; Ho *et al*., 2010]. Rather than representing sentences in a common vector space, most of these measures represent the sentences in a reduced vector space consisting only of the words contained in the sentences, and define similarity as some function of inter-sentence word-to-word similarities, where these similarities are in turn usually derived either from distributional information from some corpora (corpus-based measures), or semantic information represented in external sources such as WordNet (knowledge-based measures).

The Mihalcea *et al.* Measure

The sentence similarity measure proposed by Mihalcea *et al.* (2006) operates as follows. Given two sentences S_1 and S_2, first calculate the similarity between the first word in S_1 and each word in S_2 that belongs to the same part-of-speech class. The maximum of these scores is then weighted with the IDF score of the word from S_1. This procedure is then repeated for the remaining words in S_1, with the weighted maximum scores summed, and then normalised by dividing by the sum of IDF scores. This entire procedure is then repeated for S_2. The overall similarity is finally defined as the average of normalised weighted maximums for S_1 and S_2. In mathematical notation:

$$\begin{aligned} sim(S_1,S_2) = \frac{1}{2} & \sum_{w \in \{S_1\}} \left(\underset{x \in \{S_2\}}{\arg\max}\, sim(w,x) \times idf(w) \right) \Bigg/ \sum_{w \in \{S_1\}} idf(w) + \\ & \frac{1}{2} \sum_{w \in \{S_2\}} \left(\underset{x \in \{S_1\}}{\arg\max}\, sim(w,x) \times idf(w) \right) \Bigg/ \sum_{w \in \{S_2\}} idf(w) \end{aligned} \tag{2.5}$$

where *sim*(x, y) is the similarity between words x and y. The IDF score is determined using an external corpus. The reason for computing the semantic similarity scores only between words in the same part-of-speech class is that most WordNet-based measures are unable to calculate semantic similarity of words belonging to different parts of speech.

The Li *et al.* Measure

The sentence similarity measure proposed by Li *et al*. (2006) is based on the notion of semantic vectors. Sentences are first transformed into feature vectors having words from the sentence pair as a feature set. That is, rather than using the full set of features from some corpora, only the words appearing in the two sentences are used, thus overcoming the problem of data sparseness arising from the high dimensional vector space of a full bag of words representation. Word weights for the semantic vectors are derived from the maximum semantic similarity score between words in the feature vector and words in the corresponding sentence. If a word from the feature vector appears in the corresponding sentence, then a weight of 1 is assigned for that word; otherwise, word-to-word similarity scores are calculated between the target word (i.e., the word whose weight is being computed) and all of the words from the opposing sentence, and the weight is determined as the maximum of these similarity scores. Finally, the semantic similarity between the pair of semantic-vectors is defined as a cosine of the angle between the two vectors, as per the traditional vector-space approach. Note that Li *et al*.'s approach also utilises word order in the similarity computation, with the final similarity measure being a linear combination of semantic vector similarity and word order similarity, controlled by a mixing coefficient. This similarity measure is described further in Section 3.3.2.

These sentence similarity measures proposed by Li *et al*. (2006) and Mihalcea *et al*. (2006) have two important features in common: (i) rather than representing sentences using the full set of features from some corpora, only the words appearing in the two sentences are used, thus overcoming the problem of data sparseness arising from a full bag of words representation, and (ii) they use semantic information derived from external sources to overcome the problem of lack of word co-occurrence. Each of these measures has been used in this thesis. We now briefly describe several other recently proposed measures.

Other Sentence Similarity Measures

Islam and Inkpen (2008) proposed a corpus-based sentence similarity measure, which determines the similarity score of sentence pairs based on semantic and syntactic information (in terms of common word order) that they contain. In order to derive a fully independent sentence measure, they use three similarity functions: string similarity, word-to-word semantic similarity, and common word order similarity (used to incorporate syntactic information). Overall similarity is computed by combining these three similarity measures with normalisation. Islam and Inkpen (2008) claimed that a corpus-based measure has the advantage of large coverage when compared to a knowledge-based measure.

Ramage *et al*. (2009) introduced a variant of the vector space model based on the idea of random walks over a graph derived from WordNet, together with statistical information from a corpus. Instead of comparing vectors for each sentence directly, their method compares the distribution each sentence induces when used as the seed for a random walk over the graph. Once stationary distributions have been reached, the distributions are compared using standard vector similarity measures such as cosine similarity or Jaccard score [Manning *et al*., 2008].

Another recent contribution is from Achananuparp *et al*. (2009), who propose a novel approach that employs the semantic structure of sentences in the form of verb argument structure to measure the semantic similarity between sentence pairs. Their approach was motivated by the intuition that sentences which express the same meaning should have similar verb argument structure. Thus, instead of comparing two unstructured sentences, their method decomposes sentences into a set of verb argument roles. Given two sentences S_1 and S_2, the similarity score between the verb argument role R_1 of sentence S_1 and verb argument role R_2 of sentence S_2 is estimated by the similarity between their verbs and the sum of similarities between the corresponding arguments.

2.3.3 Word-to-Word Similarity

Each of the sentence similarity measures defined above depends in some way on a measure of semantic similarity between words. A large number of such measures have been proposed in the literature [Hindle, 1990; Hirst and St-Onge, 1998; Yang and Powers, 2005], and can broadly be categorised as either *corpus-based*, in which case similarity is calculated based on distributional information derived from large corpora, or *knowledge-based*, in which similarity is based on semantic relations expressed in external resources such as dictionaries, thesauri or WordNet. We focus on six popular knowledge-based measures defined over WordNet. These are the measures that have been used in the experiments reported in this thesis. A comprehensive review of these measures can be found in Budanitsky and Hirst (2006).

Path Measure (Rada *et al*., 1989) is the simplest of the six measures, and is based on the intuition that the shorter the path between two word-senses (i.e., synsets) in the WordNet hierarchy, the more similar they are. Thus a word is very similar to its parents or its siblings, and less similar to words that are far away in the hierarchy. Formally, it is defined as:

$$Sim_{Path}(w_i, w_j) = \frac{1}{length(w_i, w_j)} \tag{2.6}$$

where *length* is the length of the shortest path between two words (synsets) w_i and w_j, and is determined by simple node counting.

The other five measures rely on notions of lowest common subsumer (LCS) and information content (IC). Given two words (synsets) w_i and w_j in an *is-a* relation, the LCS is defined as the most specific word which both share as an ancestor. This is illustrated in Figure 2.3.

The measure proposed by Wu and Palmer (1994) computes the semantic similarity of the two words as a function of the path length from the LCS; i.e., the words' deepest common ancestor in the hierarchy:

$$Sim_{Wup}(w_i, w_j) = \frac{2 \times depth(LCS(w_i, w_j))}{depth(w_i) + depth(w_j)} \tag{2.7}$$

where *depth*(*w*) is the depth of word *w*.

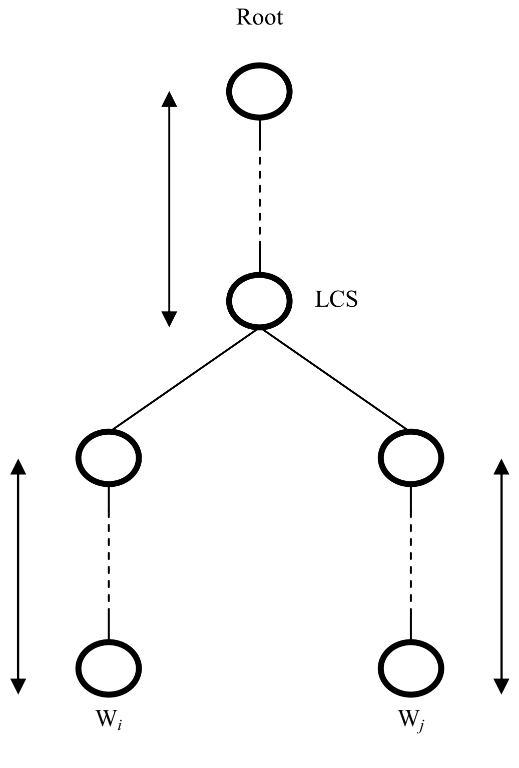

Figure 2.3: Part of a WordNet is-a hierarchy illustrating the LCS of two words.

The Resnik (1995) measure is based on the idea that the degree to which two words are similar is proportional to the amount of information they share. The measure is defined as the IC of the LCS of the two words:

$$Sim_{Res}(w_i, w_j) = IC(LCS(w_i, w_j)) \tag{2.8}$$

where $IC(w)$ is defined as $IC(w) = -\log P(w)$, where $P(w)$ is the probability that word w appears in a large corpus (e.g., the Brown corpus [Francis and Kucera, 1964]).

The Lin (1998) measure normalises the Resnik measure by dividing it by the average information content of w_i and w_j:

$$Sim_{Lin}(w_i, w_j) = \frac{2 \times IC(LCS(w_i, w_j))}{IC(w_i) + IC(w_j)}. \tag{2.9}$$

Since words at top levels have more general semantics and less similarity between them than words at lower levels, it is widely believed that better measures can be defined by taking depth into account. Leacock and Chodorow (1998) define similarity as:

$$Sim_{Lch}(w_i, w_j) = -\log \frac{N_p}{2D} \tag{2.10}$$

where N_p is the distance between the words and D is the maximum depth in the hierarchy.

The Jiang and Conrath measure (1997) is a more sophisticated measure, based on the idea that the degree to which two words are similar is proportional to the amount of information they share:

$$Sim_{J\&C}(w_i, w_j) = \frac{1}{IC(w_i) + IC(w_j) - 2 \times IC(LCS(w_i, w_j))}. \tag{2.11}$$

Several authors [Budanitsky and Hirst, 2006; Sinha and Mihalcea, 2007] have found the Jiang and Conrath measure to be superior to the five WordNet-based measures described above.

Importantly, due to the fact that WordNet organises English nouns, verbs, adjectives and adverbs separately [Harabagiu and Moldovan, 1998; Goker and Davies, 2009], all of the measures described above are only capable of calculating the similarity between words with the same part-of-speech. The problem of low connectivity in WordNet has implications for the accuracy of sentence similarity measures that utilise these word-to-word measures. This is discussed in Section 2.6.

2.3.4 Evaluation of Sentence Similarity Methods

Sentence similarity measures are commonly evaluated as *in vitro* task. To this end, a variety of sentences datasets have been constructed, and include the Microsoft Research Paraphrase (MSRP) Corpus [Dolan *et al*., 2004], and the Recognising Textual Entailment (RTE) challenge dataset [Dagain *et al*., 2005, Bar-Hair *et al*., 2006, Giampiccolo *et al*., 2007]. These datasets are discussed in Section 2.5.2. Each of these datasets is structured as a collection of sentence pairs, where each pair is tagged with a binary class value of 1 or 0, indicating whether the sentences are similar or dissimilar. The problem is thus a binary classification task in which the objective is to correctly predict the class membership of the sentence pair, with performance measured using standard binary classification measures such as accuracy, precision, recall and F-measure [van Rijsbergen, 1975]. These can be calculated from the confusion matrix shown in Figure 2.4, and are defined as follows: *Accuracy = (TP + TN)/(TP + FP + FN + TN), Recall = TP/(TP + FN), Precision = TP/(TP + FP)* and *F-measure = 2PR/(P + R),* where TP, FP, FN and TN stand for true positive, false positive, false negative and true negative, respectively.

	Actual Classification		
		+	-
Predicted Classification	+	TP	FP
	-	FN	TN

Figure 2.4: The confusion matrix for a binary classification task.

One of the difficulties in measuring the performance on *in vitro* tasks such as these is that a classification threshold must be determined, and the performance can be very sensitive to choice of this threshold. Most researchers simply use a threshold of 0.5, although some authors have experimented with other threshold values [Mihalcea *et al*., 2006; Metzler *et al*., 2007; Ramage *et al*., 2009; Achananuparp *et al*., 2009].

Unlike the binary classification datasets mentioned above, the 30-Sentence Pairs dataset [Li *et al*., 2006] (described further in Section 2.5.2) has been designed to compare the *correlation* between machine-rated similarity of sentences pairs and human-rated similarity. Importantly, testing on this dataset does not require thresholding, and in this sense the dataset can be considered as a more reliable measure of the relative similarity between sentences.

The need to determine a threshold can also be avoided by using *in vivo* evaluation model. For example, the task of clustering relies on a measure of similarity between the objects (sentences) being clustered, and the quality of clustering can be taken as being indicative of the sentence similarity measure being used. Chapter 6 of this thesis compares the performance of various sentence similarity measures *in vivo* on a number of sentence clustering tasks, and is, to the best of the author's knowledge, the first such *in vivo* evaluation of sentence similarity measures.

2.4 Sentence-Level Text Clustering

Sentence clustering plays an important role in many text-processing activities. For example, various authors have argued that incorporating sentence clustering into extractive multi-document summarisation helps avoid problems of content overlap, leading to better coverage [Hatzivassiloglou *et al.*, 2001; Radev *et al.*, 2004; Zha, 2002; Aliguyev, 2009]. However, sentence clustering can also be used within more general text mining tasks. For example consider web mining [Kosala and Blockeel, 2000], where the specific objective might be to discover some novel information from a set of documents initially retrieved in response to some query. By clustering the sentences of those documents we would intuitively expect at least one of the clusters to be closely related to the concepts described by the query terms; however, other clusters may contain information pertaining to the query in some way hitherto unknown to us, and in such a case we would have successfully mined new information.

Irrespective of the specific task (e.g., summarisation, text mining, etc.), most documents will contain interrelated topics or themes, and many sentences will be related to some degree to a number of these. This means that ideally a clustering algorithm should be able to identify soft or *fuzzy* clusters, in which sentences belong to all of these clusters with different degrees of membership. However, clustering text at the sentence level poses specific challenges not present when clustering larger segments of text, such as documents. This section first highlights some important differences between clustering at these two levels, and then examines some existing approaches to fuzzy clustering.

2.4.1 Sentence-Clustering versus Document-Clustering

Clustering text at the document level is well-established in the information retrieval literature, where documents are typically represented as data points in a high-dimensional vector space in which each dimension corresponds to a unique keyword [Salton, 1989], leading to a rectangular representation in which rows represent documents and columns represent attributes of those documents (e.g., tf-idf values of the keywords). This type of data (referred to as ‘attribute data’) is amenable to clustering by a large range of algorithms. Since data points lie in a metric space, one can readily apply prototype-based algorithms such as *k*-Means [MacQueen, 1967], Isodata [Ball and Hall, 1967], Fuzzy *c*-Means [Dunn, 1973; Bezdek, 1981] and the closely related mixture model approach [Duda *et al*., 2001], all of which represent clusters in terms of parameters such as means and covariances, and therefore assume a common metric input space. Since pairwise similarities or dissimilarities between data points can readily be calculated from the attribute data using similarity measures such as cosine similarity, one can also apply relational clustering algorithms such as Spectral Clustering [Luxburg, 2007] and Affinity Propagation [Frey and Dueck, 2007], which take as input data in the form of a square matrix $S = \{s_{ij}\}$ (often referred to as the *affinity matrix*), where s_{ij} is the (pairwise) relationship between the *i*th and *j*th data object. To distinguish it from attribute data, this type of data will be referred to as ‘relational data’. A broad range of hierarchical clustering algorithms [Theodoridis and Koutroumbas, 2008] can also be applied.

Although methods for measuring text similarity have been in existence for decades, most approaches are based on word co-occurrence. The assumption here is that the more similar two texts are, the more words they have in common. While this assumption is generally valid for large-size text fragments (e.g., documents)—and hence the widespread and successful use of these methods in information retrieval—the assumption does not hold for small-sized text fragments such as sentences, since two sentences may be semantically related despite having few, if any, words in common. To solve this problem, a number of sentence similarity measures have been proposed [Li *et al*., 2006; Mihalcea *et al*., 2006]. These measures are based on representing the sentences in a reduced vector space consisting only of the words appearing in the sentences (i.e., do not represent sentences in a common metric space), and this means that the existing prototype-based clustering algorithms such as described above, which assume a metric space, are generally not applicable for sentence clustering task. Therefore only relational clustering algorithms are applicable. The topic of interest, therefore is *fuzzy relational clustering*, i.e., fuzzy clustering based on (pairwise) relational input data.

2.4.2 Algorithms for Relational Clustering

Interestingly, the notion of fuzzy partitioning based on relational data is not new, and can be traced to the late nineteen sixties—approximately the same time as which the prototype-based *k*-Means and Isodata algorithms were first introduced. Ruspini (1969) proposed an optimisation scheme based on iteratively minimising an objective function based on pairwise dissimilarity data [Ruspini, 1969; Ruspini, 1970]. This general approach was then followed by Rouben's (1978) Fuzzy Nonmetric Model (FNM) [Roubens, 1978] and Windham's (1985) Assignment Prototype (AP) model [Windham, 1985]. However, much of this early work concentrated on defining objective functions, and in practice the results often suffered from instability in the optimisation algorithms that were used [Yang, 1993].

The first successful fuzzy relational clustering model is generally considered to be Hathaway *et al.*'s (1989) Relational Fuzzy *c*-Means (RFCM) algorithm [Hathaway *et al.*, 1989]. However, RFCM is a variant of Fuzzy *c*-Means (FCM), and is implicitly based on the notion of prototype. Thus, while RFCM operates on relational data input, it still requires that the relation expressed by this data be Euclidean (i.e., it assumes that there exists a set of data points in some space such that the squared Euclidean distance between points in this space match those in the dissimilarity relation). Non-Euclidean relations can be transformed into Euclidean ones by a transformation that adds a positive number β to all off-diagonal elements of the dissimilarity matrix, but the problem is to determine an appropriate value for β such that this Euclidean condition is met without leading to excessive loss of cluster information [Hathaway *et al*., 1989; Hathaway *et al*., 1994].

Despite its success, the Euclidean requirement in RFCM was considered restrictive, and various alternatives have been proposed. For example, the ARCA algorithm [Corsini *et al*., 2005] uses an attribute-based representation in which an object is represented by a vector of its relationships with other objects in the dataset. Thus, while the algorithm still takes relational data as input, it treats each row of the relational input matrix as a data object, thus allowing standard Fuzzy *c*-Means to be applied. Prototypes in this system are therefore objects (not necessarily present in the original dataset) whose relationship with all objects in the dataset is representative of the mutual relationships of a group of similar objects. A limitation of this approach is the high-dimensionality introduced by representing objects in terms of their similarity with all other objects.

The *k*-Medoid family of algorithms are based on the observation that in *k*-Means (and most other prototype-based algorithms), the only step that involves calculating Euclidean distances is the minimisation step, in which cluster means and covariances are updated [Hastie, 2001]. By restricting prototypes to being data points, *k*-Medoid algorithms avoid the need to calculate distances, since all calculations can be performed on the basis of pairwise relations. This idea forms the basis of the Partitioning Around Medoids (PAM) algorithm [Kaufman and Rousseeuw, 1987; Kaufman and Rousseeuw, 1990], which performs crisp clustering. Fuzzy versions of *k*-Medoids have also been proposed [Krishnapuram *et al*., 1999; Geweniger *et al*., 2010].

Like *k*-Means, methods based on *k*-Medoids are highly sensitive to the initial (random) selection of centroids, and in practice it is often necessary to run the algorithm several times from different initialisations. To overcome these problems, Frey and Dueck (2007) proposed Affinity Propagation, a technique which simultaneously considers all data points as potential centroids (or *exemplars*). Treating each data point as a node in a network, Affinity Propagation recursively transmits real-valued messages along the edges of the network until a good set of exemplars (and corresponding clusters) emerges. These messages are then updated using simple formulas that minimise an energy function based on a probability model. Frey and Dueck (2007) have shown how Affinity Propagation can be applied to the problem of extracting representative sentences from text. A fuzzy variant of Affinity Propagation was recently proposed in Gewiniger *et al*. (2009).

The family of Spectral Clustering algorithms [Luxburg, 2007], which have become very popular over the last decade, are based on matrix decomposition techniques. Data points are mapped onto the space defined by the eigenvectors associated with the top eigenvalues of the affinity matrix, and clustering is then performed in this transformed space, typically using a *k*-Means algorithm. Various spectral clustering algorithms have been proposed [Shi and Malik, 2000; Meila and Shi, 2001; Ng *et al*., 2001; Yu and Shi, 2003]. Spectral clustering has been applied to sentence clustering by Zha (2002), and Wang *et al.* (2008) have more recently applied an extended version of non-negative matrix factorisation [Lee and Seung, 2001] (which they show to be equivalent to spectral clustering) to sentence clustering in the context of multi-document summarisation.

Since Spectral Clustering uses *k*-Means in the final step, in principle it should be possible to replace the use of *k*-Means with Fuzzy *c*-Means, thereby resulting in a *fuzzy* spectral clustering algorithm. However, while Spectral Clustering generally performs very well on crisp clustering problems, there is no guarantee that applying a Gaussian model within the space defined by the eigenvalues of the affinity matrix would result in reliable fuzzy membership values. There is also the problem of introducing an extra parameter; i.e., the weighting exponent *r* that controls the fuzziness of the resulting clusters. One of the contributions of this thesis is a fuzzy relational clustering algorithm that abandons any use of explicit density models (e.g., Gaussian) for representing clusters. Instead, a graph representation is used, in which nodes represent objects, and weighted edges represent the similarity between objects (i.e. sentences). By applying the PageRank algorithm to each cluster, and interpreting the PageRank score of an object within some cluster as a likelihood, the Expectation-Maximisation (EM) framework is then used to determine the model parameters (i.e., cluster membership values and mixing coefficients). The algorithm is described in Chapter 6.

2.5 Benchmark Datasets for Experimentation

Experiments were conducted in this thesis using datasets that have been widely used in many text processing activities. These datasets are used in two main experiments: WSD evaluation, and sentence similarity measurement evaluation.

2.5.1 Datasets for Word Sense Disambiguation Experiments

Standard datasets widely used to evaluate WSD methods on *in vitro* tasks are the SemCor [Miller *et al*., 1993], Senseval-2 [Palmer *et al*., 2001] and Senseval-3 [Snyder and Palmer, 2004] datasets.

SemCor

SemCor is the largest freely-available textual corpus of syntactically and semantically annotated words, and has been extensively used in evaluating WSD systems. It consists of 352 documents that were organised into 186 documents composed of open-class (i.e., nouns, verbs, adjectives, and adverbs) sense-annotated words, and 166 documents where only verbs have been sense–annotated. The text-fragments included in SemCor were extracted from the Brown Corpus [Francis and Kucera, 1964], and each word in the text was then associated with its corresponding WordNet sense. The main purpose of this corpus is to provide instances of senses in textual context.

Overall, SemCor contains a sample of around 234,000 syntactically and semantically annotated words, thus constituting the largest sense-assigned corpus for evaluating WSD methods, both supervised and unsupervised. An excerpt of a text-fragment in this corpus is shown in Table A.1. For example, *interest* is annotated in the fourth text-fragment with part-of-speech noun and sense number 1, defined in WordNet as "*a sense of concern with and curiosity about someone or something*", compared, e.g., to sense number 4 with the same part-of-speech class defined as "*a fixed charge for borrowing money*". The original SemCor was annotated according to WordNet version 1.5. However, mappings exist to more recent versions (e.g., 2.0, 3.0, etc.). In this thesis, the SemCor corpus mapped to WordNet version 3.0 has been used.

Senseval-2 and Senseval-3

Many aspects of WSD evaluation have been standardised by the Senseval workshops [Gale *et al*., 1992; Resnik and Yarowsky, 1997; Palmer *et al*., 2006; Kilgarriff and Palmer, 2000]. This workshop provides a shared task with training and testing corpora along with sense inventories for all-words and lexical sample tasks in a variety of languages. In all-words tasks, WSD methods are used to disambiguate all open-class words in a text fragment. Unsupervised methods are typically employed in this setting, as they are applied to classify a set of unlabeled texts (i.e., test corpus). On the other hand, in lexical sample tasks, WSD methods are required to disambiguate a pre-selected set of target words, usually appearing one per text fragment. Supervised methods are often employed to handle this task as they can be trained using a number of labeled text fragments (i.e., training corpus), and then used to classify a set of unlabeled text fragments.

For the English all-words corpus in Senseval-2 and Senseval-3, nouns, verbs, adjectives and adverbs in three given texts (two from Wall Street Journal [Charniak *et al*., 2000] and one from The Brown Corpus) were manually annotated (by human annotators) using the WordNet sense to create a gold standard. This resulted in 2473 and 2081 target words, where 2239 and 1851 from those words are polysemous for Senseval-2 and Senseval-3, respectively. The three texts represent three distinct topics: news story, fiction and editorial. They were selected from the Penn TreeBank II. Human annotators were asked to annotate words in these corpora with multiple senses when WordNet contains an appropriate entry, but were asked to pick a single sense whenever possible. The annotators were also asked to mark a sample “U” when the correct sense of a word did not exist in WordNet. An excerpt of a labeled text in these corpora is reported in Table A.2. For more information on the Senseval workshops, we recommend the works of Martinez (2004) and Palmer *et al*. (2006).

2.5.2 Datasets for Sentence Similarity Experiments

Benchmark datasets that have been used in evaluating sentence similarity measures *in vitro* include the Microsoft Research Paraphrase (MSRP) [Dolan *et al*., 2004], Recognising Textual Entailment (RTE) challenge [Dagain *et al*., 2005, Bar-Hair *et al*., 2006, Giampiccolo *et al*., 2007], and 30-Sentence Pairs [Li *et al*., 2006; Li *et al*., 2009] datasets.

Microsoft Research Paraphrase (MSRP)

The MSRP dataset consists of 5801 pairs of text fragments that were automatically collected from a large number of newswire posting on the web over a period of 18 months. Each pair of text fragments was manually labeled by two human annotators with a binary true or false value, indicating whether or not the two fragments in a pair were considered a paraphrase (i.e., semantically equivalent) of each other. The agreement between the human judges was estimated at approximately 83%, which can be considered as an upper bound measure for an automatic paraphrase recognition task performed on this corpus. Table A.3 shows an excerpt of MSRP corpus. The MSRP corpus is unbalanced in that 67% of the pairs are positive (true) pairs and only

33% are negative (false). The corpus has been arbitrarily split into a training set containing 4076 text fragment pairs and a test set containing 1725 text fragment pairs. The corpus is a natural evaluation test-bed for measures of semantic similarity, and has been used by Mihalcea *et al*. (2006), Islam and Inkpen (2008), Ramage *et al*. (2009) and Achananuparp *et al*. (2009) and many others.

Recognising Textual Entailment (RTE)

Textual entailment has been introduced as a relation between text fragments, capturing the fact that the meaning of one fragment can be entailed from the other [Dagan and Glickman, 2004]. In this thesis, the test sets from second and third (RTE-2 and RTE-3) PASCAL RTE challenge have been used. Each of these datasets contains 800 pairs of *text* (T) and *hypothesis* (H) for which to determine entailment. The text-hypothesis pairs were collected by human assessors and can be decomposed into four subsets corresponding to the application domains: information retrieval, multi-document summarisation, question answering and information extraction. Similarity judgment between text-fragment pairs is based on directional inference between text and hypothesis. If the hypothesis can be inferred by the text, then that pair is considered to be a positive instance. For example the meaning of H: "*Accardo won the Paganini Competition in Genoa*" is inferred from T: "*In Accardo won the Geneva Competition and in became the first prize winner of the Paganini Competition in Genoa*", so the text entails the hypothesis. An excerpt of RTE test set is reported in Table A.4. Additional information on the RTE datasets can be found in Riabinin (2007).

30-Sentence Pairs

This dataset is due to Li *et al*. (2006), and was created by taking a set of 65 noun pairs from Rubenstein and Goodenough (1965) and replaced them with their definitions from the Collins Cobuild dictionary [Sinclair, 2001]. Cobuild dictionary definitions are written in full sentences, using vocabulary and grammatical structures that occur naturally with the word being explained. The 32 human participants were asked to complete a questionnaire, rating the similarity of meaning of the sentence pairs on the scale from 0.0 (minimum similarity) to 4.0 (maximum similarity), as in Rubenstein and Goodenough (1965). Each of the 65 sentence pairs was assigned a semantic similarity score calculated as the mean of the judgments made by the participants. When the similarity scores were averaged, the distribution of the scores was heavily skewed toward the low similarity end of the scale, with 46 pairs rated from 0.0 to 0.9, and 19 pairs rated from 1.0 to 4.0. To obtain a more even distribution across the similarity range, a subset of 30 sentence pairs was selected, consisting of all 19 sentence pairs rated 1.0 to 4.0, and 11 taken at equally spaced intervals from the 46 pairs rated 0.0 to 0.9 [Li *et al*., 2006]. Unlike the sentence similarity measure dataset described above (e.g., MSRP and RTE), in which the task is binary classification, this dataset has been used to compare correlation with human-rated similarity. Detailed information can be found in O'Shea *et al*. (2009).

2.6 WordNet

WordNet [Fellbaum, 1998] is a large structured lexical database of English words inspired by current psycholinguistic theories of human lexical memory. Unlike a traditional dictionary, which organises words alphabetically, WordNet organises English nouns, verbs, adjectives and adverbs semantically into sets of synonyms called *synsets*, each representing a single distinct concept or word sense. Thus, the synset can be viewed as a set of *synonym-words*, all expressing (approximately) the same meaning. For example, the first synset (i.e., the synset corresponding to the first sense) of the noun *car* is {car, auto, automobile, machine, motorcar}. Associated with each synset is a short written definition, referred to as its *gloss*. For example, the gloss for the above synset is "*a motor vehicle with four wheels; usually propelled by an internal combustion engine*". Each synset in WordNet is associated with its part-of-speech which we denote with a subscript: *n* for noun, *v* for verb, *c* for adjective, *r* for adverb. We use superscript to denote the sense number associated with each word. Thus, the synset described above is denoted as car_n^1. Table 2.1 shows how WordNet organises the information for a word 'bank', which has ten noun senses and eight verb senses.

Table 2.1: WordNet information for word 'bank'.

Word	Synsets (Synonyms)	Glosses
Senses(Bank$_n$):		
$bank_n^1$	{'bank'}	*"Sloping land (especially the slope beside a body of water)."*
$bank_n^2$	{'depository_financial_institution', 'bank', 'banking_concern', 'banking_company'}	*"A financial institution that accepts deposits and channels the money into lending activities."*
$bank_n^3$	{'bank'}	*"A long ridge or pile."*
$bank_n^4$	{'bank'}	*"An arrangement of similar objects in a row or in tiers."*
$bank_n^5$	{'bank'}	*"A supply or stock held in reserve for future use (especially in emergencies)."*
$bank_n^6$	{'bank'}	*"The funds held by a gambling house or the dealer in some gambling games."*
$bank_n^7$	{'bank', 'cant', 'camber'}	*"A slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force."*
$bank_n^8$	{'savings_bank', 'coin_bank', 'money_box', 'bank'}	*"A container (usually with a slot in the top) for keeping money at home."*
$bank_n^9$	{'bank', 'bank_building'}	*"A building in which the business of banking transacted."*
$bank_n^{10}$	{'bank'}	*"A flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)."*

Senses(Bank$_v$):		
$bank_v^1$	{'bank'}	*"Tip laterally."*
$bank_v^2$	{'bank'}	*"Enclose with a bank."*
$bank_v^3$	{'bank'}	*"Do business with a bank or keep an account at a bank."*
$bank_v^4$	{'bank'}	*"Act as the banker in a game or in gambling."*
$bank_v^5$	{'bank'}	*"Be in the banking business."*
$bank_v^6$	{'deposit', 'bank'}	*"Put into a bank account."*
$bank_v^7$	{'bank'}	*"Cover with ashes so to control the rate of burning."*
$bank_v^8$	{'trust', 'swear', 'rely', 'bank'}	*"Have confidence or faith in."*

A word in WordNet can either have multiple senses (i.e., polysemous) or a single sense (i.e., monosemous). For example, the noun ‘virus’ is polysemous since its three senses (biological virus, bigotry virus and computer virus) are somewhat different from each other. On the other hand, the noun ‘bigotry’ has only one sense and therefore appears in only one synset. Note that WordNet does not distinguish between polysemous and monosemous words, and also does not indicate whether two senses of a word are related to each other. In this thesis, WordNet version 3.0, which contains 155,287 words organised in 117,659 synsets for total of 206,941 word sense pairs, has been used. Statistical information regarding WordNet can be found in Table 2.2.

The synsets for each word sense are ranked according to their frequency of occurrence in the SemCor corpus, which is a subset of the Brown corpus annotated with word senses. Thus, the first sense given for a word in WordNet is attested more times in SemCor corpus than the second one, which in turn, is more frequent than the third one, etc. According to Zipf [1949], words that occur frequently in text are more polysemous than words that rarely occur.

Table 2.2: The number of words, synsets and average length (%) of glosses for each part of speech in WordNet 3.0.

Part-of-speech	Words	Synsets	Average Length of Glosses (in words)
Noun	117,798	82,115	11.1
Verb	11,529	13,767	6.2
Adjective	21,479	18,156	7.0
Adverb	4,481	3,621	4.9

2.6.1 Lexical and Semantic Relations

WordNet defines a variety of semantic and lexical relations between words and synsets. Semantic relations define a relationship between two synsets. For example, the synset {car, auto, automobile, machine, motorcar} is related to the noun synset {vehicle} through the semantic relation hypernymy, since a 'car' is a type of a 'vehicle'. Lexical relations on the other hand define a relationship between word pairs within two synsets. Thus whereas a semantic relation between two synsets relates all the words in one of the synsets to all the words in the other synset, a lexical relationship exists only between particular words of two synsets. For example the *synonymy* relation relates the words collection$_n^1$ and aggregation$_n^2$ but not the rest of the words in their respective synsets which are {collection$_n^1$, aggregation$_n^1$, accumulation$_n^2$, assemblage$_n^3$} and {collection$_n^4$, collecting$_n^1$, assembling$_n^1$, aggregation$_n^2$}. In WordNet, synonymy is considered the most important lexical relationship, and it has a special role, with synsets forming the basic units of the WordNet hierarchy.

The four most common WordNet relations for nouns are those of *hyponymy*, *hypernymy*, *holonymy*, *meronymy* and *attribute*. Hyponymy and hypernymy are semantic relationships that connect two synsets if the entity referred to by one *is a kind of* the entity referred to by the other. For example, synset {motorcycle, bike} is a kind of synset {vehicle}, therefore, synset {motorcycle, bike} is the hyponym of {vehicle}, and {vehicle} is the hypernym of {motorcycle, bike}. Holonymy and meronymy are also semantic relationships that connect two synsets if the entity referred to by one *is a part of* the entity referred to be the other. For example, if synset {sister, sis} is a part of {family, household, house, home, menage}, then synset {sister, sis} is a meronym of synset {family, household, house, home, menage}, and {family, household, house, home, menage} is a holonym of {sister, sis}. The attribute relation is a semantic relationship that connects a noun synset and an attribute (i.e., adjective) synset if the last synset is a value of noun synset. For example, the noun synset {car, auto, automobile, machine, motorcar} is related to the adjective synset {fast}, since 'fast' is attributing of 'car'. Note that this is one of the few relationships in WordNet that links synsets of different parts of speech. Another relation defined for nouns is *antonymy*. It is a lexical relationship that connects two nouns which are opposites of each other. This relation is defined between the words and not between the synsets in which those words appear. For example, the noun 'dispersion' is the antonym of the noun 'accumulation'.

WordNet defines two major semantic relations for verbs: *hypernymy* and *troponymy*. These relations represent a way of doing things. As an example, the verb synset {walk} is the troponym of verb synset {travel, go, move, locomote} since to *walk* is one way of *moving*. Like nouns, verbs are also connected through the lexical relationship of *antonymy* that links two verbs which are opposite to each other in meaning. Thus, the verb 'desist' which means to "stop an action" is the antonym of verb 'allow' which means to "permit an action".

Adjectives are defined in WordNet through the semantic relationship known as *similar to*. This relation connects two adjective synsets that are similar in meaning. For example, consider the two synsets: {quiet}, with gloss "*free of noise or uproar, or* *making little if any sound*"; and {silent, soundless, still}, with gloss *marked by absence of sound*. WordNet defines a *similar to* relation between these two synsets.

Adverbs have far fewer relations defined for them compared with the other parts of speech. One of the relations defined for adverbs is *pertainym*. This is a lexical relationship that connects adverbs to other adverbs and adjectives, and connects adverbs to nouns. This is one of few relations in WordNet links different parts of speech.

Since most relations in WordNet do not cross part-of-speech boundaries [Harabagiu and Moldovan, 1998], WordNet has only a limited number of connections between topically related words. For example, in WordNet 3.0 there is no link between the first sense of verb 'eat': "*take in solid food*" and the first sense of noun 'refrigerator': "*white goods in which food can be stored at low temperatures*". However, these words can be considered to be semantically related to the topic of 'food'. While existing sentence similarity measures define the similarity of two sentences as being some function of the semantic similarities between their constituent words, the low connectivity between topically related synsets that have different parts of speech does not allow these measures to capture the full semantic information that they contain.

Figure 2.5 shows an excerpt from WordNet representing a variety of lexical and semantic relations for some selected concepts. A comprehensive review can be found in Fellbaum (1998).

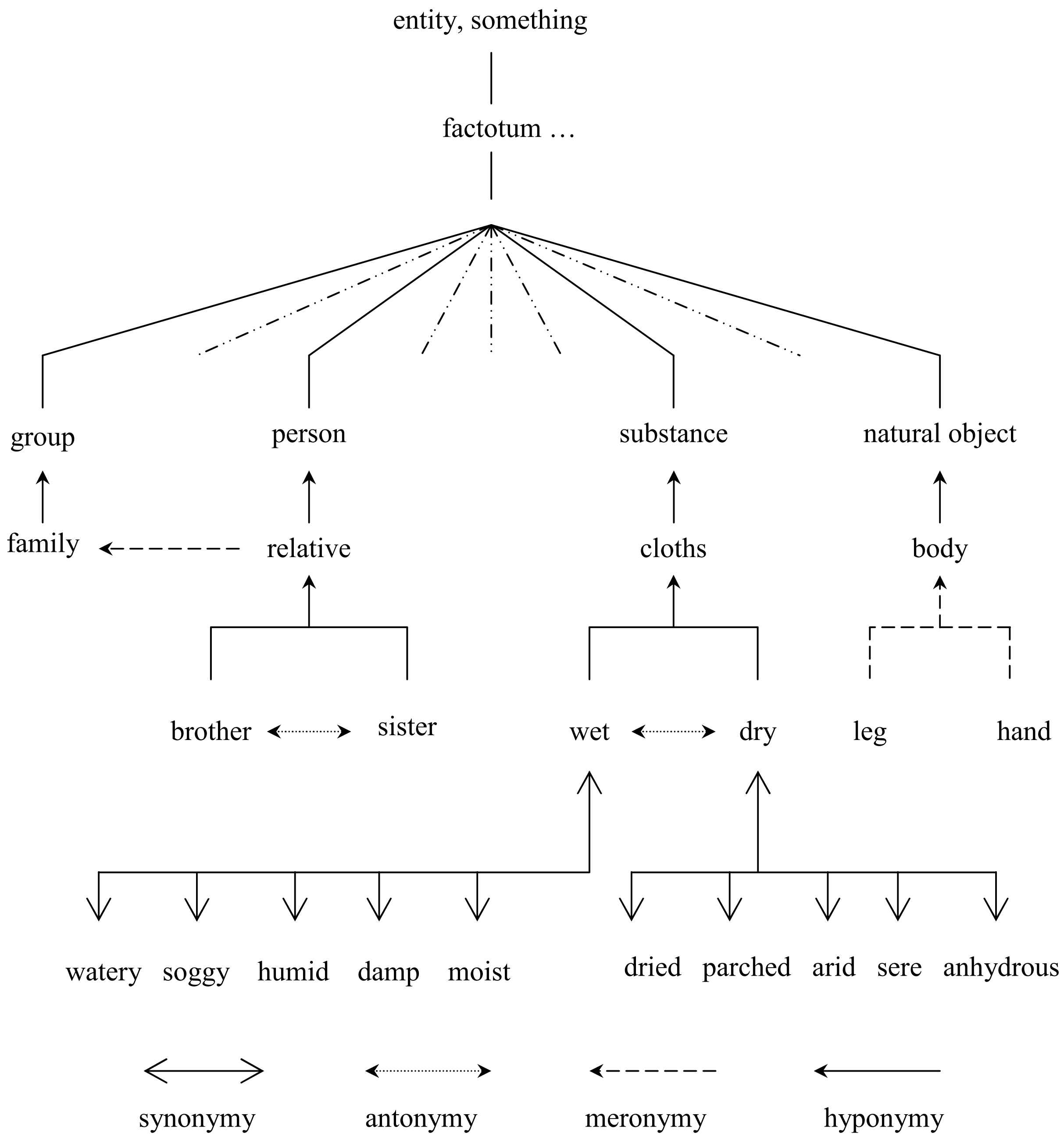

Figure 2.5: WordNet excerpt representing a variety of lexical and semantic relations.

2.7 Discussion and Concluding Remarks

This chapter has reviewed the literature and related work in the areas of word sense disambiguation, sentence similarity measurement, and sentence clustering. It is important to stress that these activities are not independent, and the performance of one often depends on the performance of another. For example, clustering sentences relies on a measure of sentence similarity; a good measure of sentence similarity should be able to effectively utilise the context; effectively using context will rely on an ability to identify the correct sense of words, and so on.

In regard to word sense disambiguation, it has been shown that graph-based methods usually achieve higher performance than their similarity-based alternatives. This is because they disambiguate all words in a text fragment simultaneously, whereas similarity-based methods disambiguate words individually, usually without considering the senses assigned to surrounding words. The main disadvantage of graphical methods is their high computational complexity. Although similarity-based methods are usually far more efficient than graph-based methods, they are usually based either on gloss-context word overlap [Lesk, 1986; Kilgarriff and Rosenzweig, 2000] or measuring pairwise similarity between word-senses [Patwardhan *et al.*, 2003], and do not fully utilise the semantic information associated with word-senses that is available through resources such as WordNet, such as glosses. Also, due to computational requirements, they are usually limited to using context from only a small window surrounding the target word. Chapter 4 proposes a new similarity-based method that has much lower complexity than graph-based methods, utilises the whole context of both surrounding words and the word-senses (glosses), yet performs comparably to, and in many cases exceeds the performance of graph-based approaches.

All of the sentence similarity measures described in this chapter compute similarity between their constituent words based either on distributional information from some corpora (corpus-based measures), or on semantic information represented in external sources such as WordNet (knowledge-based measures). Some of the measures, including those of Li *et al*. (2006) and Mihalcea *et al*. (2006), incorporate a measure based on the word's importance, as measured, for example, by the word's IDF score, or *information content*, in the case of Li *et al*. (2006). The rationale for this is that words which have a higher IDF are more important, and thus should contribute more heavily in the sentence similarity calculation than less important words. However, while it is widely accepted that incorporating IDF scores leads to improved measurement of text similarity at the document level, it is not clear that it has the same utility at the sentence level. For example, in evaluating the performance of a variety of sentence similarity measures on a range of tasks, Achananuparp *et al*. (2008) report that measures such as IDF have no clear advantage in the overall performance of these similarity measures. Another difficulty in using IDF scores at the sentence level is that many words are polysemous (i.e., have multiple meanings). Even if the sense of these words can be determined, there remains the problem that IDF scores are generally not available for specific senses of words. Chapter 3 tackles these issue by exploring the idea of incorporating into sentence similarity methods a factor based on the importance of words in the actual sentences being compared (as opposed to average importance over some large corpus).

While a number of papers have reported on the development and evaluation of sentence similarity measures [Li *et al*., 2006; Mihalcea *et al*., 2006; Metzler *et al*., 2007; Islam and Inkpen, 2008; Ramage *et al*., 2009; Achananuparp *et al*., 2009] , most of these are based on word-to-word similarity using the first sense of each of the words being compared, and intuitively, identifying the correct sense in which a word is being used should lead to a more accurate measure of the similarity between two sentences. To date there has been very little research reporting the incorporation of WSD into sentence similarity measurement. The exceptions are Abdalgader and Skabar (2010) and Ho *et al* (2010). These are described further in Chapter 5.

Another avenue for improving the measurement of sentence similarity is by incorporating more semantic information about the words in the sentences. Due to the fact that most semantic relations in WordNet do not cross part-of-speech boundaries, semantic connectivity in WordNet only exists between senses belonging to the same part-of-speech [Fellbaum, 1998] (see Section 2.6). This low connectivity between topically related senses that have different parts of speech does not allow us to fully capture the semantic information that they contain. To overcome this problem, Feng *et al*. (2008), following Zhao *et al*. (2006), have expanded the context of the sentences being compared by including WordNet relations such as synonyms, hypernyms, etc. of the words in the sentences being compared. However, the methods described by Feng *et al*. (2008) and Zhao *et al*. (2006) are based on expanding the first sense of the word, and this sense of a word in WordNet only represents the most frequently applied sense or the most general meaning [Fellbaum, 1998], but not the actual sense in which the word is being used. Intuitively, applying WSD prior to the expansion should lead to a better measure. A method which combines WSD and synonym expansion is described in Chapter 5.

In examining the literature on clustering, this chapter has focused on fuzzy relational clustering. This was motivated by the fact that: (i) widely used sentence similarity measures such as those of Li *et al*., (2006) and Mihalcea *et al*., (2006) do not represent sentences in a common metric space, thereby requiring a relational, as opposed to attribute-based, approach to clustering, and (ii) sentences are unlikely to relate to just a single concept or theme within a document, but to a number of themes. There are currently very few algorithms falling into this category. Chapter 6 presents a new fuzzy clustering algorithm which operates on relational input data; that is, data in the form of a square matrix of pairwise similarities between sentences.

Chapter 3

Sentential Word Importance

As discussed in Chapters 1 and 2, measuring similarity between sentences plays an important role in applications such as document summarisation, text mining and question answering. While various sentence similarity measures have recently been proposed [Li *et al*., 2006; Mihalcea *et al*., 2006], these measures typically only take into account word importance by virtue of inverse document frequency (IDF) scaling. IDF values are based on global information compiled over a large corpus of documents, and we hypothesise that at the sentence level better performance can be achieved by using a measure of the importance of a word within the sentence that it appears. This chapter explores the idea of incorporating into sentence similarity measurement a factor based on the importance of words in the sentences being compared. This importance will be referred to as *sentential word importance* (SWI) to distinguish it from measures such as IDF, which are derived from large corpora. Specifically, the chapter shows how the PageRank [Brin and Page, 1998] graph centrality algorithm can be used to assign a numerical measure of importance to each word in a sentence, and how these values can be incorporated within two well-known sentence similarity measures. Results show that incorporating sentential word importance leads to improvement in similarity measurement performance, as evaluated using benchmark datasets. The main content of this chapter has been presented in Skabar and Abdalgader (2010).

The chapter is structured as follows. Section 3.1 describes the use of PageRank as a measure of graph centrality. Section 3.2 then demonstrates how PageRank can be used to determine the importance of a word within the sentence that it appears. Section 3.3 outlines the sentence similarity measures due to Li *et al*. (2006) and Mihalcea *et al*., (2006) and Section 3.4 describes how word importance can be incorporated into these measures. Section 3.5 provides empirical results and Section 3.6 contains discussion and conclusions.

3.1 Measuring Graph Centrality

Various graph-based ranking algorithms have been proposed in the literature, and include eigenvector centrality [Bonacich, 1972; Bonacich, 2007], Hypertext Induced Topic Selection (HITS) [Kleinberg, 1999], PageRank [Brin and Page, 1998], Indegree, Closeness and Betweenness [Freeman, 1979]. These algorithms can be used to rank nodes (i.e., vertices) according to their centrality (or importance) in the graph. We focus here only on PageRank, as it was previously found successful in a number of applications including Web link analysis, social networks, and more recently in several text processing applications [Mihalcea *et al*., 2004].

The PageRank algorithm was originally developed to rank Web pages on the Internet. It is a variant of the eigenvector centrality measure [Bonacich, 1972; Bonacich, 2007; Brandes and Erlebach, 2005], in which the importance of a vertex within a graph can be determined by taking into account global information recursively computed from the entire graph (rather than depending only on local information about a specific vertex), with connections to high-scoring vertices contributing more to the score of a vertex than connections to low-scoring vertices. It is this importance that can then be used as a measure of centrality.

Let $G = (V, E)$ be a directed graph with the set of vertices V and set of edges E, where E is a subset of $V \times V$. PageRank assigns to every vertex in a graph a numerical score between 0 and 1, known as its *PageRank score* (*PR*). This score is defined as:

$$PR(V_i) = (1-d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j) \tag{3.1}$$

where $In(V_i)$ is the set of vertices that point to V_i, $Out(V_j)$ is the set of vertices pointed to by V_j, and d is a *damping factor*, typically set to around 0.8 to 0.9 [Brin and Page, 1998]. Using the analogy of a random surfer on the Web, nodes visited more often will be those with many links coming in from other frequently visited nodes, and the role of d is to reserve some probability for jumping to any node in the graph, thereby preventing the user getting stuck in some disconnected part of the graph.

Although originally proposed in the context of ranking Web pages, PageRank can be used more generally to determine the importance (or *centrality*) of an object in a network. For example, Mihalcea and Tarau's (2004) 'TextRank' and Erkan and Radev's (2004) 'LexRank' both use PageRank for ranking sentences for the purpose of extractive text summarisation. The underlying assumption for calculating the importance of a sentence is that sentences which are similar to a large number of other important sentences are central. Thus, by ranking sentences according to their centrality, the top-ranking sentences can then be extracted for the summary.

In both TextRank and LexRank, each sentence in a document or documents is represented by a vertex on a graph. However, unlike a Web graph, in which edges are unweighted, edges on a document graph are weighted with a value representing the similarity between sentences. The PageRank algorithm can easily be modified to deal with weighted undirected edges, resulting in the following formulation:

$$PR(V_i) = (1-d) + d \times \sum\nolimits_{j=1}^{N} \left(w_{ji} \frac{PR(V_j)}{\sum\nolimits_{k=1}^{N} w_{jk}} \right) \tag{3.2}$$

where N is the number of vertices and w_{ij} is the similarity between V_i and V_j.

3.2 Using PageRank as a Measure of Word Importance

It is straightforward to extend this idea to representing a sentence as a graph in which vertices are words, and edge weights represent the similarity between words. Edge weights can be determined using word-to-word similarity measures such as those described in Section 2.3.3. Equation 3.2 can then be used to assign to each word a score representing the importance of that word in the sentence. As an example, consider the following two sentences:

Sentence 1: “A deaf husband and a blind wife are always a happy couple.”

Sentence 2: “The woman cries before the wedding; the man afterward.”

For clarity, Figure 3.1 shows Sentence 1 represented as a graph in which vertices are words, and edges represent the semantic similarity between them. Note that stopwords have been removed. The weights are the word-to-word similarity values calculated using the Jiang and Conrath (J&C) (1997) measure. Note that in determining sentential word importance, the first WordNet sense for each word has been used.

Applying PageRank to the graph in Figure 1 results in the following, where numbers below the words are the corresponding PageRank values.

Sentence1: $\{ deaf_n^1,\ husband_n^1,\ blind_n^1,\ wife_n^1,\ happy_n^1,\ couple_n^1 \}$
[0.1106 **0.2479** 0.1157 **0.2506** 0.16667 0.1085]

Applying the same procedure to Sentence 2 results in the following.

Sentence2: $\{ woman_n^1,\quad cries_n^1,\quad wedding_n^1,\quad man_n^1,\quad afterward_n^1 \}$

[**0.2866** 0.1229 0.1096 **0.2868** 0.2000]

The words found to be most central in Sentence 1 are ‘husband’ and ‘wife’, and the words most central in Sentence 2 are ‘woman’ and ‘man’. Both of these sentences are about marriage, and the concepts of man/woman and husband/wife are clearly related to the concept of marriage. We hypothesise that incorporating these PageRank scores into the measurement of sentence similarity will result in an improved measure.

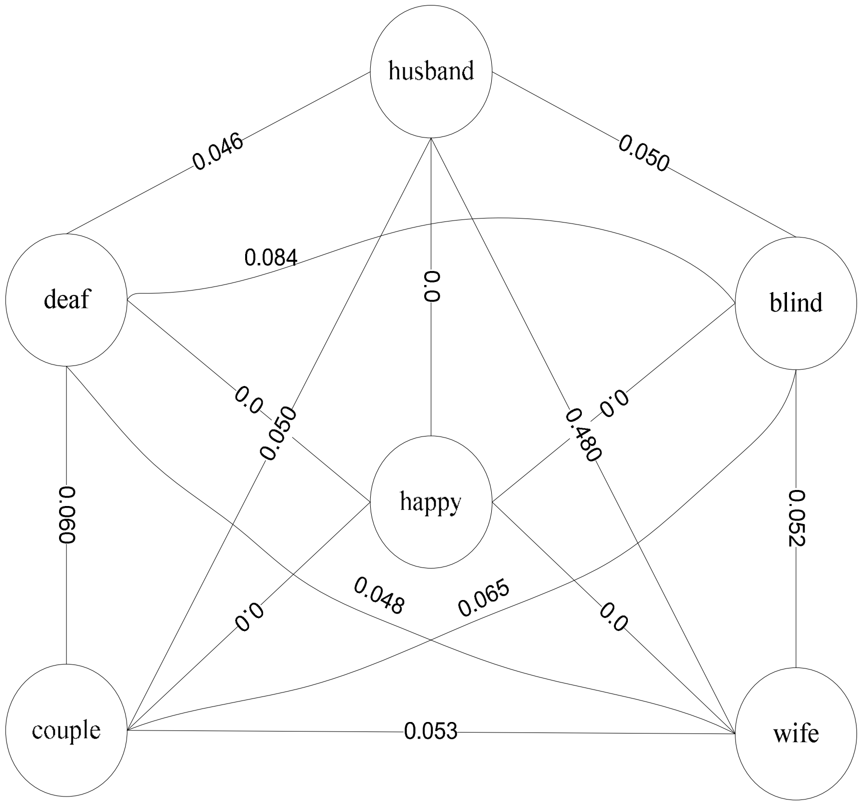

Figure 3.1: Sentence 1 represented as a graph. Edge weights represent the semantic similarity between words.

3.3 Sentence Similarity Measures

A variety of sentence similarity measures were briefly described in Section 2.3. This section describes the Mihalcea *et al.* (2006) and Li *et al.* (2006) measures in further detail. Section 3.4 will describe how these measures can be improved through incorporation of sentential word importance.

3.3.1 Mihalcea *et al.*'s Sentence Similarity Measure

The sentence similarity measure proposed in Mihalcea *et al.* (2006) computes similarity between two sentences $S_1 = \{w_{11}, w_{12}, ..., w_{1n_1}\}$ and $S_2 = \{w_{21}, w_{22}, ..., w_{2n_2}\}$, where n_i is the number of words in S_i (i = 1, 2), according to:

$$\begin{aligned} sim(S_1,S_2) = \frac{1}{2}\sum_{w\in\{S_1\}}\left(\arg\max_{x\in\{S_2\}} sim(w,x)\times idf(w)\right)\Big/\sum_{w\in\{S_1\}} idf(w) + \\ \frac{1}{2}\sum_{w\in\{S_2\}}\left(\arg\max_{x\in\{S_1\}} sim(w,x)\times idf(w)\right)\Big/\sum_{w\in\{S_2\}} idf(w) \end{aligned} \quad (3.3)$$

The computation begins by calculating the similarity score between the first word in S_2 and each word in S_1 that belongs to the same part-of-speech class. The maximum of these scores is then weighted with the *idf* score of the word from S_2. This procedure is then repeated for the remaining words in S_2, with the weighted maximum scores summed, and then normalised by dividing by the sum of *idf* scores for words in S_1. This procedure is then repeated for S_1. The overall similarity is defined as the average of normalised weighted maximums for S_1 and S_2. For convenience, we will henceforth refer to this measure simply as the *Mihalcea measure*.

3.3.2 Li *et al.*'s Sentence Similarity Measure

Whereas Mihalcea *et al.*'s (2006) approach does not utilise any explicit sentence representation, the approach proposed by Li *et al.* (2006) implicitly represents sentences in a reduced vector space. Rather than using a common vector space representation for all sentences, the two sentences being compared are represented in a reduced vector space of dimension N_u, where N_u is the number of distinct words in the union ($U = S_1 \cup S_2 = \{w_1, w_2, ..., w_{N_u}\}$, $N_u \leq n_1 + n_2$) of the two sentences. Semantic vectors, $\mathbf{V}_1$ and $\mathbf{V}_2$, are first constructed. These vectors represent sentences S_1 and S_2 in the reduced space. The similarity between S_1 and S_2 is then defined as the cosine similarity between $\mathbf{V}_1$ and $\mathbf{V}_2$. The elements of $\mathbf{V}_i$ are determined as follows. Let v_{ij} be the j^{th} element of $\mathbf{V}_i$, and let w_j be the word corresponding to dimension j in the reduced vector space. There are two cases to consider, depending on whether w_j appears in S_i:

Case 1: If w_j appears in S_i, set v_{ij} equal to 1.

Case 2: If w_j does not appear in S_i, calculate a word-word semantic similarity score between w_j and each non-stopwords in S_i, and set v_{ij} to the highest of these similarity scores i.e., $v_{ij} = \arg\max_{x \in \{S_i\}} sim(w_j, x)$.

Note that in their formulation, Li *et al.* (2006) also factor in an information content weighting so that the similarity between two words is defined as $sim(w_i, w_j) \times I(w_i) \times I(w_j)$, where $sim(w_i, w_j)$ is defined as above, and $I(w)$ is the information content of word w, and is defined as $-\log p(w) / \log(N+1)$ where $p(w)$ is the probability that the word appears in a large corpus and N is the total number of words in the corpus. As a measure of word importance, information content, therefore, plays a similar role to IDF.

Li *et al.* (2006) also utilise word order in the similarity computation, with the final similarity measure being a linear combination of semantic vector similarity and word order similarity, controlled by a mixing coefficient. While we acknowledge that incorporating word order can lead to improvements on some sentence comparison tasks, we prefer a more general measure that allows for the fact that similar meaning can be expressed not only using different combinations of words, but also with sentences that differ markedly in their structure. Therefore we do not take into account either word order or information content weighting. Note that throughout this thesis, this variation of the Li *et al.* method will be referred to as the *basic Li measure*.

3.4 Modified Sentence Similarity Measures

Incorporating PageRank values into the measures described above is relatively straightforward. The measure proposed by Mihalcea *et al.* (2006) can be modified as follows:

$$\begin{aligned} sim(S_1,S_2) = &\frac{1}{2}\sum_{w\in\{S_1\}}\left(sim\left(w, \underset{x\in\{S_2\}}{\arg\max}\left(sim(w,x)\times PR_x^{S_2}\right)\right)\times PR_w^{S_1}\right) \Bigg/ \sum_{w\in\{S_1\}} PR_w^{S_1} \;+ \\ &\frac{1}{2}\sum_{w\in\{S_2\}}\left(sim\left(w, \underset{x\in\{S_1\}}{\arg\max}\left(sim(w,x)\times PR_x^{S_1}\right)\right)\times PR_w^{S_2}\right) \Bigg/ \sum_{w\in\{S_2\}} PR_w^{S_2} \end{aligned} \tag{3.4}$$

where PR_x^S is the PageRank score of word x in sentence S. Note that this incorporates the PageRank of both the target word (i.e., words appearing in the outer summations), as well as the PageRank values of the words against which the target words are being compared.

For the basic Li measure, the only modification required is in determining the components of the semantic vectors. This can be done as follows:

Case 1: If w_j appears in S_i, set v_{ij} equal to $PR_{w_j}^{S_i}$ (i.e., the PageRank score for w_j in S_i).

Case 2: If w_j does not appear in S_i, set v_{ij} equal to the highest weighted similarity between w_j and the words in S_i; i.e., $v_{ij} = \arg\max_{x \in \{S_i\}} \left(sim(w_j, x) \times PR_{w_j}^{S_i} \right)$.

3.5 Evaluation and Experimental Results

The two similarity measures described above have been applied to the Microsoft Research Paraphrase dataset (MSRP) [Dolan *et al.*, 2004], and the Recognising Textual Entailment Challenge dataset (RTE2, RTE3) [Dagain *et al.*, 2005, Bar-Hair *et al.*, 2006, Giampiccolo *et al.*, 2007], both of which have become benchmark datasets used in evaluating sentence similarity measures. These datasets have been described in Section 2.5.2.

3.5.1 Paraphrase Recognition

Since the MSRP dataset is a binary classification task, a classification threshold needs to be determined (i.e., the candidate pair is classified as a paraphrase if the similarity score exceeds this threshold). The results presented in this section are based on a 0.5 threshold, as is common practice in the literature [Mihalcea *et al.*, 2006; Metzler *et al.*, 2007; Ramage *et al.*, 2009]. We experiment with other thresholds in a later section. Classification performance is evaluated in terms of accuracy, precision, recall and F-measure, which were described in Section 2.3.4.

Table 3.1 shows the performance of the Li measure on the MSRP datasets using the six word-to-word semantic similarity measures described in Section 2.3.2. The first section of the table shows performance with the use of sentential word importance; the second shows performance without sentential word importance (i.e., basic Li measure, in which word-to-word similarity is measured based on the use of first WordNet sense of the component words). When sentential word importance is applied, the best performance in terms of overall accuracy and F-measure was achieved using the J&C measure (Accuracy = 72.5%, F-measure = 82.4%), followed closely by the Path measure (Accuracy = 71.0%, F-measure = 81.8%). Without the use of sentential word importance, the accuracy and F-measure achieved using the J&C measure drop to 68.9% and 80.5% respectively, and the accuracy and F-measure due to the Path measure drop to 68.6% and 80.5% respectively. While the incorporation of sentential word importance leads to improved performance when used in conjunction with the J&C and Path measures, it has no discernable effect when used in conjunction with the other four word-to-word similarity measures. Similar conclusions can be drawn from Tables 3.2, which shows results obtained using the modified Mihalcea measure.

Table 3.1: Performance (%) of SWI-modified and basic Li measure on MSRP dataset (classification threshold = 0.5).

Measure	Accuracy	Precision	Recall	F-measure
SWI-modified basic Li measure				
J&C	**72.5**	71.5	97.2	**82.4**
Path	**71.0**	70.2	97.9	**81.8**
Lch	66.6	66.5	100.0	79.9
Resnik	66.5	66.5	100.0	79.9
Lin	67.4	67.3	99.0	80.1
Wup	66.4	66.4	99.9	79.8
Basic Li measure				
J&C	68.9	69.0	96.8	80.5
Path	68.6	68.5	97.7	80.5
Lch	66.6	66.5	100.0	79.9
Resnik	66.4	66.4	100.0	79.9
Lin	67.6	67.3	99.1	80.1
Wup	66.4	66.4	100.0	79.8

Table 3.2: Performance (%) of Mihalcea and SWI-modified Mihalcea measure on MSRP dataset (classification threshold = 0.5).

Measure	**Accuracy**	**Precision**	**Recall**	**F-measure**
SWI-modified Mihalcea measure				
J&C	**71.3**	71.2	95.5	**81.5**
Path	**71.0**	70.9	95.2	**81.5**
Lch	66.6	66.5	100.0	79.9
Resnik	66.5	66.5	100.0	79.9
Lin	68.4	68.4	97.6	80.4
Wup	66.5	66.6	99.5	79.8
Mihalcea measure				
J&C	69.5	69.9	94.9	80.5
Path	69.7	70.0	95.2	80.7
Lch	66.6	66.5	100.0	79.9
Resnik	66.5	66.5	99.9	79.8
Lin	68.6	68.5	97.5	80.5
Wup	66.7	66.7	99.6	79.9

The low precision and high recall in the results shown in Tables 3.1 and 3.2 suggest that the 0.5 classification threshold results in a small number of false negatives (paraphrases classified as non-paraphrases) at the expense of a large number of false positives (non-paraphrases classified as paraphrases). To determine whether better performance can be obtained using some other threshold, the two SWI-modified measures were applied using eleven different similarity thresholds ranging from 0.0 to 1.0 with interval 0.1. Figure 3.2 shows how the accuracy varies with the classification threshold for sentence similarity calculated using the J&C word-to-word measure. It can be seen from the figure that the optimal threshold for the SWI-modified Mihalcea method is 0.6, whereas the optimal threshold for the SWI-modified Li measure is 0.5. Similar results were obtained using the Path measure (not shown). Table 3.3 shows full details of the performance for these optimal thresholds.

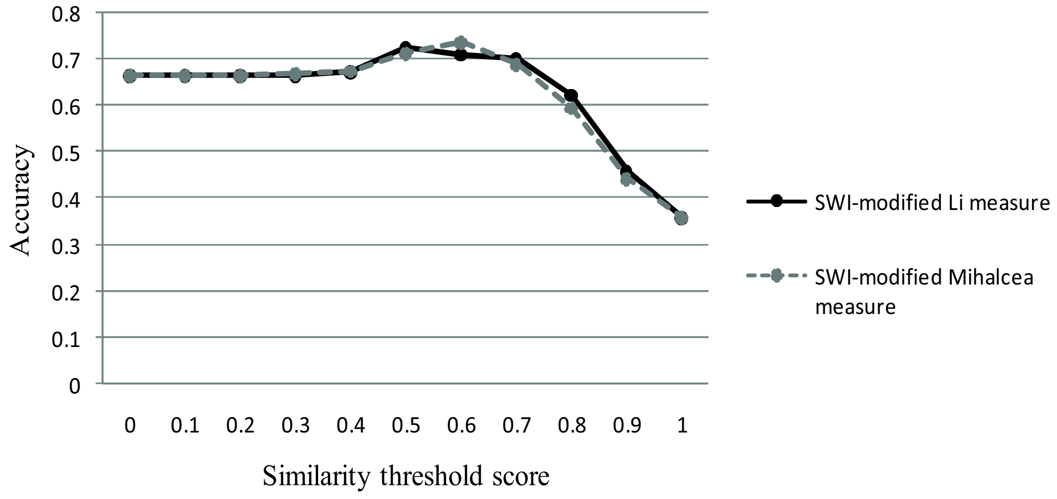

Figure 3.2: Accuracy vs. classification threshold curve of SWI-modified basic Li and Mihalcea measures.

Table 3.3: Performance corresponding to optimal thresholds for SWI-modified basic Li, and SWI-modified Mihalcea measures using J&C measure.

Measure	Threshold	Accuracy	Precision	Recall	F-measure
		SWI-modified basic Li measure			
J&C	0.5	**72.5**	71.5	97.2	82.4
Path	0.5	71.0	70.2	97.9	81.8
		SWI-modified Mihalcea measure			
J&C	0.6	**73.7**	76.6	86.9	81.4
Path	0.6	73.3	76.1	87.4	81.3

As expected, increasing the threshold for the SWI-modified Mihalcea method has resulted in a better balance between Precision and Recall; however, the optimal threshold for the SWI-modified basic Li method remained at 0.5. It is, of course, possible that better performance could be achieved by exploring more fine-grained thresholds (e.g., by incrementing thresholds by 0.01 instead of 0.1). However, there is little point in doing this for practical tasks, since tuning the threshold would require use of a training set, and as mentioned earlier, we are only interested in unsupervised classification. Hence we do not perform any such tuning.

Table 3.4: Performance (%) of other approaches and baselines on MSRP dataset (classification threshold is 0.5, except for Islam and Inkpen's method in which threshold is 0.6).

Measure	Accuracy	Precision	Recall	F-measure
Mihalcea *et al.* (2006), Corpus-based				
PMI-IR	69.9	70.2	95.2	81.0
LSA	68.4	69.7	95.2	80.5
Mihalcea *et al.* (2006), WordNet-based				
L&C	69.5	72.4	87.0	79.0
J&C	69.3	72.2	87.1	79.0
Resnik	69.0	69.0	96.4	80.4
Ramage *et al.*, (2009) Random Graph Walk				
Cosine	68.7	-	-	78.7
Dice	70.8	-	-	80.1
JS	68.8	-	-	80.5
Islam and Inkpen (2008), Corpus-based				
STS	72.6	74.7	89.1	81.3
Baselines				
Vector-based	65.4	71.6	79.5	75.3
Random	51.3	68.3	50.0	57.8

For comparative purposes Table 3.4 shows the performance of some other sentence similarity measures that have been reported in the literature: Mihalcea *et al.*'s (2006) corpus-based and WordNet-based measures; the random graph walk method of Ramage *et al.* (2009) using three distributional similarity measures; Islam and Inkpen's (2008) corpus-based measure, and, following Mihalcea *et al.* (2006), a baseline that measures the cosine similarity between vectors in a full bag of words representation with *tf-idf* weighting, and a random baseline, created by randomly assigning a true or false value to pairs of text fragments. The reported classification threshold in each case was 0.5, with the exception of Islam and Inkpen (2008), who use a 0.6 threshold.

The accuracies of 72.5% and 71.3% achieved on this dataset respectively by the SWI-modified basic Li measure (with classification threshold = 0.5) and the SWI-modified Mihalcea measure (with classification threshold = 0.5) using the J&C word similarity measure are higher than that of any of the methods shown in Table 3.4. (We have not tested statistical significance of these results). The accuracy of 73.7% achieved using the SWI-modified Mihalcea method with a classification threshold of 0.6 exceeds the performance of all methods shown in Table 3.4. We note, however, that the papers reporting the results of many of the methods shown in Table 3.4 used a default 0.5 classification threshold, and better performance might have been achieved using a different classification threshold.

Correlation Analysis

Many applications which might require sentence similarity to be measured are in fact not classification problems, and thus do not require thresholding. Ultimately it is relative—not absolute—similarity values which are important. Sentence clustering is an example. For these tasks, it is not so important that different measures yield values similar in magnitude. What is more important is that the values are highly correlated. To this end, Table 3.5 shows the Pearson correlation between similarity values obtained for the MSRP dataset using the SWI-modified basic Li measure. It is interesting to note that the word-to-word measures which result in the most highly correlated values (J&C and Path) are precisely those which were found to give the best performance when used with SWI in Table 3.3. Note that the J&C and Path measures have a higher correlation with the other knowledge-based measures, which is probably due to the data-driven information content used in those measures, although they have an additional normalisation factor that makes them behave differently. A more detailed description of word-to-word similarity measures is provided in Section 2.3.3.

Table 3.5: MSRP dataset similarity measure correlations.

	Path	Wup	Lch	Lin	Resnik	**J&C**
Path	1.000	0.874	0.653	0.951	0.741	**0.993**
Wup		1.000	0.745	0.927	0.928	0.848
Lch			1.000	0.690	0.819	0.601
Lin				1.000	0.819	0.942
Resnik					1.000	0.702
J&C						1.000

3.5.2 Textual Entailment Recognition

As discussed in Chapter 2, textual entailment recognition is the task of determining whether a text fragment is entailed by a hypothesis (another text fragment). Entailment is an asymmetric relation based on directional inference, and symmetric similarity measures such as the Li measure and Mihalcea measure should not generally be expected to perform as well as measures designed to utilise a deeper semantic analysis specifically formulated to determine entailment. Nevertheless, the dataset has been previously been used as a measure of (asymmetric) sentence similarity, and we follow suit.

Table 3.6 shows the results of applying the SWI-modified basic Li similarity measure to recognising textual entailment (RTE2 and RTE3 test datasets). The value in the column labelled ‘threshold’ is the optimal threshold determined by incrementing the threshold in steps of 0.1. The results clearly indicate that the use of SWI leads to improved performance, as determined using all four evaluation measures. The same observations can be made for the SWI-modified Mihalcea method, as shown in Table 3.7.

Table 3.6: Performance (%) of SWI-modified basic Li measure on RTE2 and RTE3 test datasets (optimal classification threshold = 0.5).

Data	Measure	Threshold	Accuracy	Precision	Recall	F-measure
			SWI-modified basic Li measure			
RTE2-test	J&C	0.5	**64.2**	63.0	65.3	**64.1**
	Path	0.5	**64.6**	62.1	70.9	**66.2**
			Basic Li measure			
RTE2-test	J&C	0.5	56.1	54.9	57.9	56.3
	Path	0.5	56.2	54.4	65.3	59.3
			SWI-Modified basic Li measure			
RTE3-test	J&C	0.5	**65.3**	66.4	63.2	**64.8**
	Path	0.5	**67.0**	67.3	66.9	**67.1**
			Basic Li measure			
RTE3-test	J&C	0.5	61.1	61.9	59.3	60.5
	Path	0.5	60.7	60.6	63.0	61.8

Table 3.7: Performance (%) of SWI-modified Mihalcea measure on RTE2 and RTE3 test datasets (optimal classification threshold = 0.5).

Data	Measure	Threshold	Accuracy	Precision	Recall	F-measure
			SWI-modified Mihalcea measure			
RTE2-test	J&C	0.5	**64.3**	62.5	68.1	**65.2**
	Path	0.5	**64.8**	62.6	70.1	**66.1**
			Mihalcea measure			
RTE2-test	J&C	0.5	56.8	55.3	61.7	58.3
	Path	0.5	57.8	56.1	63.7	59.7
			SWI-modified Mihalcea measure			
RTE3-test	J&C	0.5	**67.0**	67.4	66.7	**67.0**
	Path	0.5	**67.8**	68.1	67.9	**68.0**
			Mihalcea measure			
RTE3-test	J&C	0.5	61.5	61.7	61.7	61.7
	Path	0.5	62.0	62.2	62.2	62.2

3.6 Discussion and Conclusions

The idea of incorporating word importance in text similarity measurement is not new, and IR researchers have been using IDF weights in measuring document similarity for decades. However, IDF weights are determined using an external corpus, and while this may provide information on how important a word is when taken over a large corpus, it provides little information on the importance of a word in the context of the sentence in which it appears. This chapter has described how the PageRank algorithm can be used to determine sentential word importance, and how the resulting importance scores can then be incorporated into two well-known sentence similarity measures. Results show that incorporating sentential word importance leads to improvement in sentence similarity measurement, as evaluated using two benchmark datasets.

In regard to increased computational cost, the step of calculating sentential word importance contributes extremely little to the calculation. Graph-based centrality measures such as PageRank converge quickly, even for a relatively large number of nodes. Assuming that sentences contain in the order of ten or so words, any increase in computational cost is virtually insignificant when compared, for example, against the cost of computationally expensive tasks such as IDF scaling over a large corpus of documents, unless IDF might be counted once and stored in advance for further use.

The performance of sentence similarity measures such as those described above depend on the word-to-word similarity measure that is employed. Word-to-word similarity measures vary from simple edge-counting, to more sophisticated approaches which attempt to factor in peculiarities of the WordNet hierarchy by discovering link direction (depth and path) and density, with different access to sources such as statistical and knowledge corpora. Improved word-to-word similarity measurement would be expected to translate directly into improved sentence similarity measurement.

Most sentence similarity measures, including those of both Li *et al*., (2006) and Mihalcea *et al*., (2006) define the similarity of two sentences as being some function of the semantic similarities between their constituent words. However, many words have more than one meaning (polysemy), and in order to accurately calculate the similarity between two sentences it is therefore important to correctly identify the sense in which the words are being used in those sentences. In calculating word-to-word similarities, the Li *et al*., (2006) and Mihalcea *et al*., (2006) measures both assume the first WordNet sense of the word (i.e., synset), which from the shortest path between the two words, and intuitively it should be possible to achieve better performance by correctly determining the correct sense. The next chapter presents a new algorithm for WSD. Chapter 5 then shows how the WSD technique can be incorporated within a novel sentence similarity measure.

Chapter 4

Similarity-Based WSD using Context Vectors

Word sense disambiguation (WSD)—the process of identifying the actual meanings of words in a given text fragment—has a long history in the field of computational linguistics. Due to its importance in understanding the semantics of natural language, it is considered one of the most challenging problems facing this field. This chapter proposes a new unsupervised similarity-based WSD algorithm that operates by computing the semantic similarity between glosses of the target word and a context vector. The sense of the target word is determined as that for which the similarity between gloss and context vector is greatest. The chapter also shows how performance can be further improved by incorporating a preliminary step in which the relative importance of words within the original text fragment is estimated, thereby providing an ordering that can be used to determine the sequence in which words should be disambiguated. Empirical results show that the proposed method performs favourably against the state-of-the-art unsupervised WSD methods, as evaluated on several benchmark datasets.

The chapter is structured as follows. Section 4.1 describes the proposed WSD method. Section 4.2 presents a walk-through example demonstrating how the method operates. Empirical results are presented in Section 4.3, and Section 4.4 provides discussion and conclusions.

4.1 Proposed Algorithm

The WSD algorithm described in this section has previously been reported in Abdalgader and Skabar (2011). Unlike current approaches, which are based either on gloss-context word overlap [Lesk 1986, Kilgarriff and Rosenzweig 2000] or measuring pairwise similarity between words [Patwardhan *et al*., 2003; Sinha and Mihalcea, 2007; Navigli and Lapata, 2010], the proposed method determines the sense of a target word by measuring the semantic similarity between its WordNet glosses and the context provided by all remaining words in the given text fragment, which will be referred to as the 'context vector'. The correct sense of the target word is identified as the sense for which the semantic similarity between gloss vector and context vector is highest. Importantly, whereas conventional unsupervised WSD methods are based on measuring pairwise similarity between *words*, the proposed approach is based on measuring semantic similarity between *sentences*. This enables it to utilise a higher degree of semantic information, and is more consistent with the way that human beings disambiguate; that is, by considering the greater context in which the word appears.

The basic algorithm will first be presented, which will be referred to as *Word Sense Disambiguation using Context Vectors* (WSDCV). It will then be shown how its performance can be further improved by incorporating a preliminary step in which the relative importance of words within the original text fragment is estimated, these estimates then being used to provide an ordering that can be used to determine the sequence in which words should be disambiguated. Disambiguated words can then progressively be used in the disambiguation of remaining words.

4.1.1 Basic Algorithm

The text fragment (e.g., sentence) containing the words to be disambiguated is first represented as the set of non-stopwords that it contains:

$$W = \{w_i \mid i = 1..N\} \tag{4.1}$$

where N is the number of words in W. Stopwords (e.g., a, and, or, but, etc.) are removed because they do not carry any semantic information. Suppose that w_i is the target polysemous word whose sense need to be disambiguated, and G_{w_i} represents the set of WordNet glosses corresponding to the various senses of w_i i.e.:

$$G_{w_i} = \left\{g_{w_i}^{k} \mid k = 1..N_{w_i}\right\}, \tag{4.2}$$

where N_{w_i} is the number of WordNet senses for w_i, and $g_{w_i}^{k}$ is the set of non-stopwords in the k^{th} WordNet gloss of w_i. Let R_i be the context vector comprising all words from W, except w_i:

$$R_i = \left\{w_j \mid w_j \in W \text{ and } j \neq i\right\}. \tag{4.3}$$

The sense for word w_i is identified as the k value for which $g_{w_i}^{k}$ is semantically most similar to R_i. The full procedure is described in Algorithm 1.

Algorithm 1. The *WSDCV* algorithm.

```
Input: Words $W = \{w_i \mid i = 1..N\}$
       Glosses $G_{w_i} = \left\{g_{w_i}^{k} \mid k = 1..N_{w_i}\right\},\ i = 1..N$
Output: WordNet senses $T = \{t_i \mid i = 1..N\}$ where $t_i$ is the WordNet sense of $w_i$.

Word Sense Disambiguation (WSD)

1: for $i$=1 to $N$ do
2:     $R_i = \{w_j \mid w_j \in W \text{ and } j \neq i\}$ // with considering the senses assigned to neighbouring words
3:     max_sim ← 0
```

```
4:    $t_i \leftarrow 1$
5:    **for** $j$=1 to $N_{w_i}$ **do**
6:       tmp $\leftarrow$ similarity$\left(morph\left(g_{w_i}^{j}\right), R_i\right)$ // gloss and context vectors semantic similarity
7:          **if** tmp > max_sim **then**
8:             max_sim $\leftarrow$ tmp
9:             $t_i \leftarrow j$
10:         **end if**
11:    **end for**
12: **end for**
```

Note on Morphological Processing

In line 6 of the algorithm, in which similarity between gloss and context vectors is calculated, a function *morph* has been applied to the gloss vector. This function takes a set of words as input, and returns a set of the same length consisting of the corresponding morphological stems of the base words (cf., Section 2.1.2). For example, the word 'stem' is the base form of the words 'stemmer', 'stemming' and 'stemmed'. Since the gloss is a brief description of the meaning of a word (usually without syntactic structure), it is preferable to preserve as much of the meaning of the gloss's words as possible, and this requires avoiding noise such as that arising through counting different word forms (e.g., 'car' and 'cars') as separate words. This means keeping words in their base form. However, morphological processing has not been applied to the context vector R_i. There are two related reasons for this. Firstly, we wish to take advantage of the additional information about the part-of-speech of words in the context vector. This information may be lost if morphological processing is applied. Secondly, not applying morphological processing reduces the number of glosses to be considered.

4.1.2 Gloss and Context Vectors Similarity

To compute the semantic similarity between gloss and context vectors as shown in line 6 of Algorithm 1, the basic Li measure described in Section 3.3.2 is used.

Let W_1 and W_2 be the word contexts of the two text fragments (sentences) whose similarity is to be calculated. Assume that W_1 is the gloss vector corresponding to the WordNet sense k for the target word w_i and W_2 is the corresponding context vector:

$$W_1 = g_{w_i}^{k} = \{w_{1i} \mid i = 1..N_1\}, \tag{4.4}$$

$$W_2 = R_i = \{w_{2i} \mid i = 1..N_2\}. \tag{4.5}$$

The union word set U is first constructed by combining all distinct words from W_1 and W_2:

$$U = W_1 \cup W_2 = \left\{ w_i \mid i = 1..N_U \le N_1 + N_2 \right\}, \tag{4.6}$$

where N_U is the total number of words in U. The union word set can be viewed as the semantic information board for the compared sentences.

Semantic vectors $\mathbf{V}_1$ and $\mathbf{V}_2$, corresponding to W_1 and W_2 respectively, are then constructed. Each entry of these vectors corresponds to a word in the union set U, so their dimension is N_U. Let v_{ij} be the j^{th} element of $\mathbf{V}_i$, and let w_j be the corresponding word from U. The value of v_{ij} is determined according to the semantic similarity of w_j to all words in W_i. The similarity between $\mathbf{V}_1$ and $\mathbf{V}_2$ is then calculated using the method described in detail in Section 3.3.2.

4.1.3 Incorporating Word Importance

The WSDCV algorithm described above disambiguates words in a left-to-right fashion. It can be improved by incorporating a preliminary step in which the sentential word importance of each target word in the text fragment is estimated, thereby providing an ordering which can be used to determine the sequence in which words should be disambiguated. Detailed description of how the PageRank algorithm [Brin and Page 1998] can be used to estimate the relative importance of words in a sentence can be found in Chapter 3 (cf., Section 3.1 and Section 3.2).

In using word importance information for WSD, there is a choice of either disambiguating less important words first, or more important words first. Intuitively, disambiguating less important words first is expected to better utilise the available semantic information. To see why, consider the most important word. If this word is disambiguated incorrectly, this is likely to have a major negative impact on subsequent use of this information. Ideally, the most important word should be disambiguated based on the already disambiguated senses of less important words, and this suggests that the disambiguation ordering should be from least important to most important, with disambiguated senses progressively incorporated into the disambiguation for as yet undisambiguated words.

Incorporating this step adds virtually no computational expense to the overall disambiguation process. This is because only a single instance of PageRank needs to be applied. Moreover, the graph to which PageRank is applied is small, since the sentence will generally contain only a small number of words.

4.1.4 Computational Complexity

Let a be the average number of senses per word, g the average length of gloss vectors, and N the number words in the sentence to be disambiguated. Each word to be disambiguated will require a calculation of similarity between each of its possible gloss vectors and the context vector (i.e., *a* calculations, on average). The total number of such calculations required to disambiguate all target words is therefore $O(N.a)$. Each of these calculations requires $O((N-1).g)$ word-to-word similarity calculations. Therefore the total number of word-word similarity calculations is $O(N.(N-1).a.g)$.

It is instructive to compare this with the complexity of Patwardhan *et al.*'s (2003) approach (see Section 2.2.1), in which the total number of word-to-word similarity calculations is $O(a^{N_w})$, where N_w is the number of words in the selected window around the target word. For illustrative purposes, suppose that *a*, *N* and *g* are all equal to 10 (i.e., 10 senses per word, 10 words per gloss, and 10 words in the sentence to be disambiguated, which are not dissimilar to that which might be expected in practice). The WSDCV method would therefore require approximately 9,000 word-word similarity calculations. The number required by Patwardhan *et al.*'s method, however, depends critically on the size of the context window. For example, if N_w = 3, then 1,000 calculations are required, but if N_w = 4, this increases to 10,000, which already exceeds the number required by the WSDCV method. What is important is that the WSDCV method uses the context provided by all remaining words for a cost that can only be achieved using Patwardhan *et al.*'s (2003) approach with a relatively small context window.

4.2 A Walk-through Example

The following example is provided to illustrate the WSDCV method described above. Consider the sentence "The virus spread in all saving deposit money systems in the bank", which contains the polysemous words 'virus' and 'bank'. First construct the set *W,* which contains all non-stopwords:

$$W = \{\text{'virus', 'spread', 'saving', 'deposit', 'money', 'systems', 'bank'}\}$$

Then calculate the importance of each word using the method of sentential word importance described in Chapter 3, resulting in the following PageRank scores.

$$PageRank_{\text{Scores}} = [0.1350,\ 0.1438,\ 0.1401,\ 0.1366,\ 0.1452,\ 0.1577,\ 0.1412]$$

Based on these scores, order the words in *W* sequentially from lowest to highest scores, resulting in:

W = ('virus', 'deposit', 'saving', 'bank', 'spread', 'money', 'systems')

Now, after the words have been ordered based on their importance scores, the basic WSDCV algorithm starts by disambiguating the first word, which is *virus* (least important word). This word has three WordNet glosses:

virus^1_n : (virology) ultramicroscopic infectious agent that replicates itself only within cells of …

virus^2_n : a harmful or corrupting agency.

virus^3_n : a software program capable of reproducing itself and usually capable of causing …

Construct the contexts g^1_{virus}, g^2_{virus} and g^3_{virus} corresponding to each of these glosses. For space reasons, only the second is shown:

g^2_{virus} = {'harmful', 'corrupt', 'agency'}

Since the *virus* is being disambiguated, the context vector R_{virus} will consist of all words from *W* except *virus*:

R_{virus} = ('deposit', 'saving', 'bank', 'spread', 'money', 'systems')

The union set is formed by combining words from g^2_{virus} and R_{virus}:

U = {'deposit', 'saving', 'bank', 'spread', 'money', 'systems', 'harmful', 'corrupt', 'agency'}

Note that as described in Section 4.1.1, in constructing the union set morphological processing has been applied to the words from W_1, as per line 6 of the word sense disambiguation algorithm, but not to the context vector W_2 (i.e., R_{virus}).

Now compute the semantic vectors $\mathbf{V}_1$ and $\mathbf{V}_2$, corresponding to g^2_{virus} and R_{virus} respectively (as described in Section 4.1.2), resulting in:

$$\mathbf{V}_1 = [0.071, 0.083, 0.076, 0.076, 0.083, 0.071, 1.0, 1.0, 1.0]$$

$$\mathbf{V}_2 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.083]$$

Finally, calculating the cosine of these vectors gives a similarity score of 0.128.

Applying the same procedure to the first and third WordNet glosses of *virus*, results in similarity scores of 0.201 and 0.243 respectively. Thus, the sense of word *virus* in the original sentence will be determined as that corresponding to the third WordNet sense, which is clearly the sense in which *virus* would be interpreted by a human in this context.

Repeating this same procedure to the remaining words in *W* results in the following:

Target Word	**Context Vectors** (R_i)
virus	('deposit', 1), ('saving', 1), ('bank', 1), ('spread', 1), ('money', 1), ('systems', 1)
deposit	('saving', 1), ('bank', 1), ('spread', 1), ('money', 1), ('systems', 1), **('virus', 3)**
saving	('bank', 1), ('spread', 1), ('money', 1), ('systems', 1), **('virus', 3), ('deposit', 4)**
bank	('spread', 1), ('money', 1), ('systems', 1), **('virus', 3), ('deposit', 4), ('saving', 5)**
spread	('money', 1), ('systems', 1), **('virus', 3), ('deposit', 4), ('saving', 5), ('bank', 8)**
money	('systems', 1), **('virus', 3), ('deposit', 4), ('saving', 5), ('bank', 8), ('spread', 10)**
systems	**('virus', 3), ('deposit', 4), ('saving', 5), ('bank', 8), ('spread', 10), ('money', 1)**

{('virus', 3), ('deposit', 4), ('saving', 5), ('bank', 8), ('spread', 10), ('money', 1), ('systems', 1)}

where the numbers indicate the sense of the associated word, and boldface indicates how senses assigned to disambiguated words are considered progressively during disambiguation of the remaining words. The glosses of these senses, together with their part-of-speech, are as follows:

virus$^{3}_{n}$: a software program capable of reproducing itself and usually capable of causing great ...

deposit$^{4}_{n}$: money deposited in a bank or some similar institution.

saving$^{5}_{v}$: accumulate money for future use.

bank$^{8}_{n}$: a container (usually with a slot in the top) for keeping money at home.

spread$^{10}_{v}$: distribute over a surface in a layer.

money$^{1}_{n}$: the most common medium of exchange; functions as legal tender.

systems$^{1}_{n}$: instrumentality that combines interrelated interacting artifacts designed to work …

Note that some of these senses are different to what a human would assign (e.g., a human would identify the word 'saving' as an adjective, not a verb); however, given the available range of WordNet senses, the assignments are quite reasonable.

4.3 Evaluation and Experimental Results

This section presents results of evaluating the effectiveness of the WSDCV method as *in vitro* using three benchmark datasets: the SemCor [Miller *et al.*, 1993], Senseval-2 [Palmer *et al.*, 2001], and Senseval-3 [Snyder and Palmer 2004] datasets. These datasets have been described in detail in Section 2.5.1.

4.3.1 SemCor

SemCor is the largest common textual corpus of semantically annotated words, and has been extensively used in evaluating WSD systems. It consists of 352 documents that were organised into 186 documents composed of open-class sense-annotated words, and 166 documents where only verbs have been sense-annotated.

Results on the SemCor corpus are shown in Table 4.1, where the numbers in the table represent the accuracy (i.e., the number of correct true prediction senses divided by the total number of true prediction senses). The first section of the table shows performance with the use of sentential word importance; the second shows performance with the use of sentential word importance under a different word ordering strategy (highest to lowest importance); the third shows performance without use of sentential word importance (i.e., words disambiguated in a left-to-right fashion). When word importance is applied, the best performance in terms of overall accuracy was 53.10% achieved using the Path measure, followed closely by an accuracy of 53.01% achieved using J&C (Jiang and Conrath) measure. Without the use of word importance, the accuracy achieved using the Path and J&C measures drops to 50.60% and 51.01% respectively, demonstrating that the incorporation of sentential word importance (from lowest to highest importance order) does lead to improved performance. It can also be seen that using word importance in reverse order does not lead to any improvement over left-to-right ordering.

Table 4.1: WSDCV Performance (% accuracy) on SemCor corpus using various disambiguation ordering and various word-to-word similarity measures.

Part-of-Speech	**Path**	**J&C**	**Res**	**Lin**	**Wup**	**Lch**
WSDCV with Sentential Word Importance, lowest to highest ordering						
All	**53.10**	**53.01**	43.60	51.08	49.04	48.03
WSDCV with Sentential Word Importance, highest to lowest ordering						
	50.30	51.90	41.20	50.01	48.80	47.75
WSDCV without Sentential Word Importance, left-to-right ordering						
All	50.60	51.01	44.90	49.60	48.01	47.51

For comparative purposes, Table 4.2 shows the performance of the graph-based method reported in Navigli and Lapata (2010), as well as three baselines: First-Sense, Random and ExtLesk. The First-Sense baseline is determined by assigning the first WordNet sense (i.e., the most-frequent sense, MFS) of the word being disambiguated. This method is generally considered an upper bound, and it is prevalent for all-words methods (i.e., methods that disambiguate all words, irrespective of their part-of-speech) to perform below this baseline, since word senses in WordNet are ordered from most-frequent to least-frequent [Fellbaum, 1998]. The Random baseline is established by randomly assigning senses to words, and can be considered a lower bound. The ExtLesk baseline corresponds to the extended glosses method [Banerjee and Pedersen, 2003], originally introduced in Lesk (1986). As can be seen from Table 4.2, the performance of WSDCV exceeds those of both the Random and ExtLesk baselines, as well as the results reported in Navigli and Lapata (2010) for both local and global measures. Note, however, that the performance of Navigli and Lapata's method could be improved by using more semantic relation edges.

Table 4.2: Performance (% accuracy) of other WSD methods and baselines on the SemCor corpus.

Part-of-Speech	**Measure**		
Navigli and Lapata (2010) Graph-based, *Local*			
All	Degree	PageRank	HITS
	50.01	49.76	44.29
Navigli and Lapata (2010) Graph-based, *Global*			
All	Compactness	Graph Entropy	Edge Density
	43.53	42.98	43.54
Baselines			
All	First-Sense (MFS)	Random	ExtLesk
	74.17	39.13	47.85

4.3.2 Senseval-2 and Senseval-3 All Words

The Senseval-2 and Senseval-3 datasets are subsets of the Wall Street Journal corpus, and published as part of the Senseval WSD evaluation workshop. Each contains approximately 2000 target words, approximately 1700 of which are polysemous and annotated with WordNet senses. Senseval-2 and Senseval-3 are divided into two major categories: English all-words (where all words have been sense-annotated), and lexical sample (where only a relatively small subset of words have been sense–annotated). Since the WSDCV method is unsupervised, and operates to disambiguate all words, only the all-words dataset was used in these experiments.

Table 4.3: WSDCV Performance (% accuracy) on Seneval-2 and Senseval-3 datasets.

Measure	**Senseval-2**	**Senseval-3**
	WSDCV method	
J&C	**60.13**	**59.66**
Path	**59.57**	**58.15**
	Navigli and Lapata (2010), Graph-based	
Degree	-	52.90
	Sinha and Mihalcea (2007), Graph-based	
Combined Voting	57.57	-
	Baselines	
Best Supervised	69.00	65.20
First Sense (MFS)	58.00	62.40
ExtLesk	31.70	43.10

Performance of the WSDCV method using the Path and J&C measures is shown in the first section of Table 4.3. (The other four word-to-word measures performed significantly worse, and are not shown). The other sections of the table show results from the graph-based methods of Navigli and Lapata (2010) and Sinha and Mihalcea (2007), as well as several baselines. WSDCV outperforms the graph-based methods and the ExtLesk method on both datasets. On Senseval-2, it also slightly outperforms the First Sense (MFS) baseline. Since the Best Supervised baseline is based on supervised learning using sense-labeled data, it is of no surprise that its performance is significantly above all other methods shown.

4.4 Discussion and Conclusions

This chapter has presented a new similarity-based WSD method based on the use of context vectors. In contrast with conventional approaches, which measure similarity between word senses over a fixed-size context window around the target word, the WSDCV method measures the similarity between gloss vectors of the target word, and a context vector comprising the remaining words in the text fragment containing the words to be disambiguated. This has been motivated by the belief that human beings disambiguate words based on the whole context that contains the target words, usually under a coherent set of meanings. While WordNet does not provide anywhere close to the same level of knowledge about words that a human being has, it offers at least a portion of such information through its definitional glosses and semantic relations between words. The empirical results have shown the method to achieve excellent performance against both the state-of-the-art unsupervised WSD methods and baseline measures, as evaluated on several standard datasets.

The performance of the method depends to a large extent on the word-to-word similarity measure that is employed. Of the six knowledge-based measures (all based on WordNet synsets) which have been applied, it has been found that in general the J&C and Path measures lead to the best performance. However, the algorithm itself is orthogonal to the actual word-to-word similarity measure used, and new word-to-word similarity measures can easily be incorporated into the method, and may lead to further improvement in disambiguation performance.

Aside from the use of context vectors, another novel aspect of the WSDCV approach is the use of sentential word importance to estimate the relative importance of target words, thereby providing an ordering that can be used to determine the sequence in which words should be disambiguated. The importance of words is determined by applying PageRank to the graph composed of all words in the sentence. This adds virtually negligible computational cost, since the computation is performed only once (i.e., as a pre-processing step prior to the disambiguation process). Moreover, sentences are small, and consequently PageRank will be quick to converge. The empirical results have shown that incorporating such an ordering clearly affects the disambiguation performance, leading to improved performance over a simple left-to-right ordering. An obvious variation on this approach is to apply PageRank again after each word has been disambiguated, since the additional context provided by the disambiguated word may result in different importance values for the remaining words, thereby possibly leading to improved performance in disambiguating those words. However, the disadvantage of this is the additional computational expense incurred.

The results presented in this chapter have been based on *in vitro* evaluation tasks. However, the primary interest in this thesis lies in applying WSDCV in the context of more encompassing NLP applications (*in vivo*) such as sentence clustering. Clustering performance will obviously depend on a good sentence similarity measure, and a good sentence similarity measure may in turn be based on incorporating WSD. This is explored in the next chapter.

Chapter 5

Synonym Expansion for Sentence-Level Text Similarity Measurement

Measuring the similarity between text fragments at the sentence level is made difficult by the fact that two sentences that are semantically related may not contain any words in common. This means that standard IR measures of text similarity, which are based on word co-occurrence and designed to operate at the document level, are not appropriate. To address this problem, various sentence similarity measures have been recently proposed, and include those due to Li *et al*., (2006) and Mihalcea *et al*., (2006), which have been used in the experiments described in the previous chapters. These measures have two important features in common: (i) rather than representing sentences using the full set of features from some corpora, only the words appearing in the two sentences are used, thus overcoming the problem of data sparseness arising from a full bag-of-words representation, and (ii) they use semantic information derived from external sources to overcome the problem of lack of word co-occurrence. Intuitively, the performance of such measures depends on the amount and quality of the semantic information that is available, and the purpose of this chapter is to explore how such measures could be improved through better utilising the semantic information available from lexical resources such as WordNet [Fellbaum, 1986]. The main contribution described in the chapter is a novel sentence similarity measure which uses WSD and synonym expansion to provide a richer semantic context to measure sentence similarity.

The chapter is structured as follows. Section 5.1 introduces the proposed sentence similarity measure. Section 5.2 presents a walk-through example demonstrating how the measure operates. Empirical results are presented in Section 5.3 and Section 5.4 concludes the chapter.

5.1 Proposed Method

The sentence similarity measure described in this section has previously been reported in Abdalgader and Skabar (2010). Unlike existing measures, which use the set of exact words that appear in the sentences, the proposed method constructs an expansion word set for each sentence using synonyms of the words in that sentence. The method is depicted in Figure 5.1.

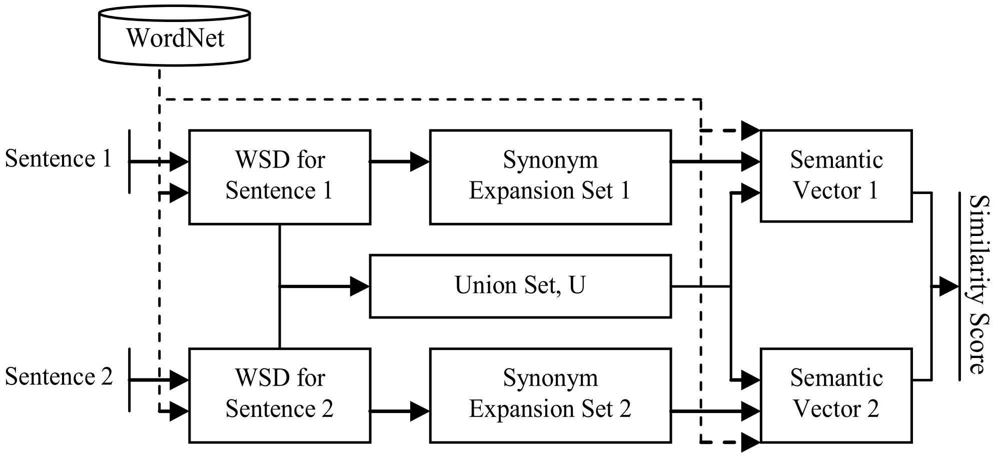

Figure 5.1: Sentence Similarity Computation Diagram.

For each of the sentences being compared, a word sense disambiguation step is first applied in order to identify the sense in which words are being used within the sentence. A synonym expansion step is then applied, resulting in a richer semantic context from which to estimate semantic vectors. The similarity between semantic vectors can then be calculated using a standard vector space similarity measure such as cosine similarity. For convenience, the method will be referred to as the *Sentence Similarity using Synonym Expansion* (SSSE) measure. Since the SSSE measure requires WSD to be applied as a preliminary step, the role of WSD will first be described.

5.1.1 Incorporating Word Sense Disambiguation

Sentence similarity as measured using methods such as those described in Section 2.3.2 is based on word-to-word similarities. The standard approach used within WordNet-based similarity measures is to simply use the first WordNet sense for each of the two words being compared (i.e., ignore to identify the actual sense of the word) [Li *et al*., 2006; Mihalcea *et al*., 2006; Zhao *et al*., 2006; Islam and Inkpen]. This is usually justified on the grounds that senses in WordNet are ordered from most-frequent to least-frequent (see Section 2.6). However this can lead to inaccurate similarity measurements. To illustrate, consider the following:

Sentence 1: "I deposited a cheque at the bank."
Sentence 2: "There is oil sediment on the south bank of the river."

Using the reduced vector space representation (cf., Section 3.3.2):

$S_1 = \{deposited_v^1, cheque_n^1, bank_n^1\}$

$S_2 = \{oil_n^1, sediment_n^1, south_n^1, bank_n^1, river_n^1\}$

$U = \{river_n^1, south_n^1, oil_n^1, sediment_n^1, deposited_v^1, cheque_n^1, bank_n^1\}$

$\mathbf{V}_1 = [0.066, 0.062, 0.058, 0.055, 1.0, 1.0, 1.0]$

$\mathbf{V}_2 = [1.0, 1.0, 1.0, 1.0, 0.0, 0.059, 1.0]$

where S_1 and S_2 contain all non-stopwords; U is the reduced vector space, consisting of all words in the union of S_1 and S_2; and $\mathbf{V}_1$ and $\mathbf{V}_2$ are the semantic vectors for S_1 and S_2 in this reduced vector space. Calculating the cosine similarity between $\mathbf{V}_1$ and $\mathbf{V}_2$, results in a value of 0.33.

The similarity value of 0.33 is likely to be an overestimate. For example, the word 'bank' appears in both sentences, but its sense is clearly different in each. Using a WordNet sense of 0 (i.e., first sense) will always result in a maximum similarity between these words, since the first sense of a word will obviously be identical with itself. Problems might also arise between words which are not common between the two sentences. For example, there is a sense of 'deposit' which is closely related to 'sediment' (an oil deposit might be considered a sediment), and it will be the similarity between these senses that is then used in the sentence similarity calculation.

Performing WSD prior to measuring sentence similarity affects the result dramatically. To demonstrate, applying WSD results in the following sense-assigned words:

$$S_1 = \{\, deposited_v^1,\ cheque_n^1,\ bank_v^4 \,\}$$

$$S_2 = \{\, oil_v^1,\ sediment_v^1,\ south_n^4,\ bank_n^1,\ river_n^1 \,\}$$

$$U = \{\, river_n^1,\ bank_n^1,\ south_n^4,\ deposited_v^2,\ sediment_v^1,\ cheque_n^1,\ oil_v^1,\ bank_v^4 \,\}$$

$$\mathbf{V}_1 = [0.059, 0.051, 0.052, 1.0, 0.044, 1.0, 0.050, 1.0]$$

$$\mathbf{V}_2 = [1.0, 1.0, 1.0, 0.050, 1.0, 0.059, 1.0, 0.049]$$

Calculating the cosine similarity between $\mathbf{V}_1$ and $\mathbf{V}_2$ in this case results in a value of 0.11. This is much lower than that achieved without the use of WSD, and is more in accord with the human judgement that S_1 and S_2 bear little semantic similarity.

Now consider the following sentences, which most humans would consider to be semantically related:

Sentence 3: “The world is in economic crisis.”

Sentence 4: “The current dismal fiscal situation is global.”

Calculating sentence similarity with and without WSD results in similarity values of 0.08 and 0.09 respectively. It is problematic that a value 0.08 has been obtained for a pair of sentences which are considered to be semantically related, yet a higher value of 0.11 was obtained for Sentences 1 and 2, which are considered to be not semantically related.

The relatively low similarity value measured between Sentences 3 and 4 can be partly explained by the fact that most semantic relations in WordNet do not cross part-of-speech boundaries. That is, in WordNet, semantic *connectivity* only exists between synsets (word senses) belonging to the same part-of-speech (cf., Section 2.6). For example in WordNet 3.0, there is no link between the noun $crisis_n^1$ and the adjective $dismal_c^1$, and thus their similarity evaluates to zero. However, these words can be considered to be semantically related. This low connectivity between topically related synsets that have different parts of speech does not allow us to capture the semantic information that they contain. The following section describes how using synonym expansion can help solve this problem.

5.1.2 Increasing Semantic Context through Synonym Expansion

Since synsets provide a means of expanding semantic context, it can be hypothesised that a better measure of sentence similarity can be achieved by incorporating synonyms. This way, one synonym word can increase the connectivity between topically related synsets. For example, consider the following actual synonyms of noun $crisis_n^1$ (number of synonyms is 1) and adjective $dismal_c^1$ (number of synonyms is 10):

$$crisis_n^1 \;=\; \{crisis_n^1\},$$

$$dismal_c^1 \;=\; \{blue_n^1,\ dingy_c^1,\ dark_n^1,\ gloomy_c^1,\ disconsolate_c^1,\ grim_c^1,\ sorry_c^1,\ drab_n^1,\ drear_c^1,\ dreary_c^1\}.$$

By comparing all synonyms of noun $crisis_n^1$ semantically with all synonyms of adjective $dismal_c^1$, as shown in Figure 5.2 (using the Jiang and Conrath similarity measure), the synonym 'crisis' of the noun $crisis_n^1$ is found to have a semantic relation with the three synonyms 'blue', 'dark' and 'drab' of the adjective $dismal_c^1$. As can be seen from the figure, the value of 0.093 is considered as the final similarity value (the highest resulting value) between noun $crisis_n^1$ and adjective $dismal_c^1$, rather than value of zero that is obtained when comparing the words directly without synonym expansion.

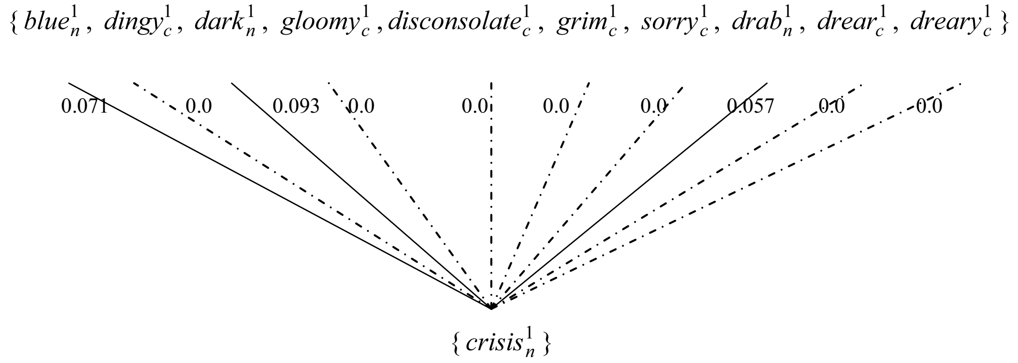
$\{blue_n^1,\ dingy_c^1,\ dark_n^1,\ gloomy_c^1, disconsolate_c^1,\ grim_c^1,\ sorry_c^1,\ drab_n^1,\ drear_c^1,\ dreary_c^1\}$
0.071
0.0
0.093
0.0
0.0
0.0
0.0
0.057
0.0
0.0
$\{crisis_n^1\}$

Figure 5.2: Increased semantic connectivity between related words. Numbers indicate similarity scores between words.

Now consider Sentences 3 and 4 above. Disambiguating the words in these sentences results in a vector space consisting of the following sense-assigned words:

$$U = \{\, fiscal_c^1,\ current_c^1,\ crisis_n^1,\ dismal_c^1,\ situation_n^1,\ global_c^1,\ world_n^2,\ economic_c^1 \,\}$$

The information from the respective synsets of these words can be used to add context to the original sentences. For example, Sentence 4 above was originally represented as the set:

$$S_4 = \{\, current_c^1,\ dismal_c^1,\ fiscal_c^1,\ situation_n^1,\ global_c^2 \,\}$$

Using information from the synsets of these words, this can be expanded to:[1]

$$\mathit{Synonym\ Expansion\ Set}_4 = \{\, current_n^1,\ blue_n^1,\ dark_n^1,\ dingy_c^1,\ disconsolate_c^1,\ gloomy_c^1,\ grim_c^1,\ sorry_c^1,\ drab_n^1,\ drear_c^1,\ dreary_c^1,\ financial_c^1,\ state\_of\_affairs_n^1,\ ball-shaped_c^1,\ globose_c^1,\ globular_c^1,\ orbicular_c^1,\ spheric_c^1,\ spherical_c^1 \,\}$$

The same procedure can now be applied to Sentence 3. This sentence was originally represented as the set:

$$S_3 = \{\, world_n^2,\ economic_c^1,\ crisis_n^1 \,\}$$

Using information from the synsets of these words, this can be expanded to:

$$\mathit{Synonym\ Expansion\ Set}_3 = \{\, domain_n^1,\ economical_c^1,\ crisis_n^1 \,\}$$

[1] Note that all synonyms are added with first sense (i.e., WordNet sense 1). We discuss this at the conclusion of the example.

Computing the semantic vectors for Sentences 3 and 4 results in the following::

V$_3$: [0.0, 0.807, 1.0, 0.0, 0.0, 0.068, 0.059, 0.0]
V$_4$: [1.0, 1.0, 0.111, 1.0, 0.0, 0.074, 1.0, 0.0]

Calculating the cosine similarity between **V**$_3$ and **V**$_4$ results in a value of 0.38, which is much higher than the value of 0.08 achieved without synonym expansion, and far more consistent with the likely judgement that a human would make.

It is important to note that all synonyms are added with first sense (i.e., WordNet sense 1). While this might appear counter-intuitive, since this may not be the sense of the synonym in the original synset (i.e., the synset of the word being expanded), it is precisely through including synonyms with sense 1 that we are able to expand the context. There are two inter-related reasons for this. Firstly, adding the correct sense for a synonym would achieve nothing, since the similarity of some word *x* to this synonym would be the same as its similarity to all other words in the same synset (which includes the identified sense of the original word used to produce the synset). Secondly, WordNet assigns a sense of 1 to the most frequently used sense of a word. This means that using this sense is most likely (but not guaranteed) to expand the context in a semantic direction of benefit in finding possible semantic similarities between words in the two sentence being compared.

Note also that the original disambiguated words have not been added to the synonym expansion sets. There are two reasons for this. Firstly, adding the original words with their first sense would loss the actual meaning of disambiguated words. For example, the disambiguated noun $world_n^2$ in S$_3$ has two synonyms ($world_n^1, domain_n^1$), so if we add the synonym $world_n^1$ to synonym expansion set$_3$, the incorrect sense for noun *world* will be considered in the similarity calculation. Secondly, adding the original disambiguated words to the synonym expansion sets would achieve the same similarity value of their synonyms. For example, noun

$situation_n^1$ in S$_4$, which has two synonyms ($situation_n^1, state\_of\_affairs_n^1$), the similarity value of synonym $situation_n^1$ and synonym $state\_of\_affairs_n^1$ is the same value with disambiguated noun $situation_n^1$ in U (i.e., since we considering the highest similarity value in semantic vectors). The same case for synonyms ($world_n^2, domain_n^1$) with disambiguated noun $world_n^2$. Moreover, use of synonym expansion does not require the dimensionality of the vector space to be increased (i.e., synonyms have not been added to U). The expanded context is utilised when calculating the semantic vectors. Whereas originally the entries for these vectors were based on similarities to words in the original sentence, similarities to the synonyms that have been introduced are now also considered.

WSD and synonym expansion pull in opposite directions: WSD tends to decrease similarity values; synonym expansion tends to increase them. Thus, even though synonym expansion has increased the similarity value for Sentences 3 and 4, it is likely also to have increased the similarity value for Sentences 1 and 2. While it may appear that WSD and synonym expansion are working at odds, this is not the case. What is crucial to note is that synonym expansion is based on identified word senses. The semantic context is not expanded blindly, but is focused in the direction of the semantic context provided by the sense-assigned meanings of the original words. Synonym expansion is not independent from WSD, it *requires* WSD.

5.2 A Walk-through Example

For clarity, a complete example of the method described in the previous section is now provided. Consider the following two sentences, which contain the polysemous words ‘virus’ and ‘bank’:

Sentence 1: "The virus spread in all saving deposit money systems in the bank."
Sentence 2: "All fish in the river of the south bank have been infected by the virus."

Removing stopwords and performing WSD results in the following sets of sense-assigned words:

$S_1 = \{ deposit_n^4,\ virus_n^3,\ saving_v^5,\ bank_n^8,\ spread_v^{10},\ money_n^1,\ systems_n^1 \}$

$S_2 = \{ virus_n^1,\ bank_n^1,\ south_n^4,\ infected_v^2,\ river_n^1,\ fish_n^1 \}$

The synonym expansion sets are then constructed using all synonyms of the sense-assigned words.

$Synonym\ Expansion\ Set_1 = \{ bank\_deposit_n^1,\ computer\_virus_n^1,\ save_n^1,\ lay\_aside_v^1,\ save\_up_v^1,\ savings\_bank_n^1,\ coin\_bank_n^1,\ money\_box_n^1,\ spread_n^1,\ money_n^1,\ systems_n^1 \}$

$Synonym\ Expansion\ Set_2 = \{ virus_n^1,\ bank_n^1,\ south_n^1,\ infect_v^1,\ taint_n^1,\ river_n^1,\ fish_n^1 \}$

Forming the union set U by combining all disambiguated words from S_1 and S_2 results in a vector space consisting of the following sense-assigned words:

$U = \{ river_n^1,\ virus_n^1,\ saving_v^5,\ spread_v^{10},\ systems_n^1,\ virus_n^3,\ south_n^4,\ infected_v^2,\ money_n^1,\ deposit_n^4,\ fish_n^1,\ bank_n^8,\ bank_n^1 \}$

Semantic vectors for synonym expansion set$_1$ (S_1) and synonym expansion set$_2$ (S_2) can be formed from U and WordNet. The procedure of deriving these vectors is shown in Table 5.1.

The first row of the Table 5.1 lists words in the union set U, the first section of the first column lists words in synonym expansion set$_1$, the second section of the first column lists words in synonym expansion set$_2$ and all words are listed in the order as they appear in union set and synonym expansion sets. For each word in U, the semantic similarity value is calculated between two words at the cross point with synonym expansion set$_1$ (synonym expansion set$_2$). Note that the value at the cross point is set to 0.0 when two words have no semantic similarity. The Jiang and Conrath (1997) measure has been used to calculate the semantic similarity between words. The semantic vectors $\mathbf{V}_1$ ($\mathbf{V}_2$) are obtained by selecting the highest value in each column, resulting in the semantic vectors:

$\mathbf{V}_1$ = [0.079, 0.067, 1.0, 0.048, 1.0, 1.0, 0.071, 0.043, 1.0, 1.0, 0.081, 1.0, 0.070]

$\mathbf{V}_2$ = [1.0, 1.0, 0.042, 0.043, 0.081, 0.052, 0.081, 0.039, 0.066, 0.053, 1.0, 0.067, 1.0]

Table 5.1: The procedure of deriving semantic vectors for synonym expansion set 1 and 2.

U	$river_n^1$	$virus_n^1$	$saving_v^5$	$spread_v^{10}$	$systems_n^1$	$virus_n^3$	$south_n^4$	$infected_v^2$	$money_n^1$	$deposit_n^4$	$fish_n^1$	$bank_n^8$	$bank_n^1$
Synonym Expansion Set$_1$													
$bank\_deposit_n^1$	0.053	0.044	0.00	0.00	0.052	0.043	0.047	0.00	0.269	1.0	0.050	0.046	0.046
$computer\_virus_n^1$	0.052	0.043	0.00	0.00	0.051	1.0	0.046	0.00	0.052	0.043	0.049	0.045	0.045
$save_n^1$	0.050	0.042	0.00	0.00	0.049	0.041	0.045	0.00	0.050	0.042	0.047	0.044	0.044
$lay\_aside_v^1$	0.00	0.00	1.0	0.048	0.00	0.00	0.00	0.043	0.00	0.00	0.00	0.00	0.00
$save\_up_v^1$	0.00	0.00	1.0	0.048	0.00	0.00	0.00	0.043	0.00	0.00	0.00	0.00	0.00
$savings\_bank_n^1$	0.046	0.039	0.00	0.00	0.046	0.039	0.042	0.00	0.046	0.039	0.044	0.041	0.041
$coin\_bank_n^1$	0.066	0.057	0.00	0.00	0.095	0.045	0.060	0.00	0.056	0.046	0.067	1.0	0.059
$money\_box_n^1$	0.066	0.057	0.00	0.00	0.095	0.045	0.060	0.00	0.056	0.046	0.067	1.0	0.059
$spread_n^1$	0.062	0.051	0.00	0.00	0.061	0.049	0.054	0.00	0.063	0.051	0.058	0.053	0.053
$money_n^1$	0.066	0.053	0.00	0.00	0.065	0.052	0.057	0.00	1.0	0.269	0.061	0.056	0.056
$systems_n^1$	0.079	0.067	0.00	0.00	1.0	0.051	0.071	0.00	0.065	0.052	0.081	0.095	0.070
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
$\mathbf{V}_1 =$	0.079	0.067	1.0	0.048	1.0	1.0	0.071	0.043	1.0	1.0	0.081	1.0	0.070
Synonym Expansion Set$_2$													
$virus_n^1$	0.062	1.0	0.00	0.00	0.067	0.043	0.057	0.00	0.053	0.052	0.070	0.057	0.056
$bank_n^1$	0.066	0.056	0.00	0.00	0.070	0.045	0.061	0.00	0.056	0.046	0.065	0.059	1.0
$south_n^1$	0.069	0.058	0.00	0.00	0.072	0.046	0.081	0.00	0.058	0.048	0.068	0.061	0.062
$infect_v^1$	0.00	0.00	0.042	0.043	0.00	0.00	0.00	0.039	0.00	0.00	0.00	0.00	0.00
$taint_n^1$	0.050	0.043	0.00	0.00	0.050	0.042	0.045	0.00	0.051	0.043	0.048	0.044	0.044
$river_n^1$	1.0	0.062	0.00	0.00	0.079	0.052	0.068	0.00	0.066	0.053	0.074	0.066	0.066
$fish_n^1$	0.074	0.070	0.00	0.00	0.081	0.049	0.067	0.00	0.061	0.050	1.0	0.067	0.065
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
$\mathbf{V}_2 =$	1.0	1.0	0.042	0.043	0.081	0.052	0.081	0.039	0.066	0.053	1.0	0.067	1.0

Calculating the cosine of these vectors using the J&C measure results in a sentence similarity score of 0.136. This value is relatively low, and indicates that the two sentences bare little semantic similarity, despite containing several common words.

5.3 Evaluation and Experimental Results

This section presents results from applying the SSSE measure to three benchmark datasets: the Microsoft Research Paraphrase (MSRP) [Dolan *et al.*, 2004], Recognizing Textual Entailment Challenge (RTE2 and RTE3) [Dagain *et al.*, 2005], and 30-Sentence Pairs [Li *et al.*, 2006] datasets. These datasets have been described in Section 2.5.2.

5.3.1 Paraphrase Recognition

Table 5.2 shows the performance of SSSE measure on the MSRP dataset. Also shown are the previously reported results from applying the SWI-modified basic Li and Mihalcea measures (from Chapter 3), as well other recently reported results and two baselines. Vector-based baseline measures cosine similarity between vectors in a full bag-of-words representation with *tf-idf* weighting. Random baseline was created by randomly assigning a true or false value to pairs of text fragments. Baselines are due to Mihalcea *et al.* (2006). As per the procedure of Chapter 3, the classification threshold was incremented in steps of 0.0, and results reported in the table correspond to the threshold giving the best performance. Note that the optimal threshold for SSSE was 0.6, whereas the optimal threshold for SWI-modified basic Li measure was 0.5. This is consistent with the comments made above regarding the effect of introducing synonym expansion; i.e., that incorporating synonym expansion will tend to increase the average similarity values, thus rasing the optimal threshold.

Table 5.2: Comparison of performance (%) of SSSE measure with other similarity measures and baselines on MSRP dataset.

Measure	Threshold	Accuracy	Precision	Recall	F-measure
		SSSE measure			
J&C	0.6	**74.6**	75.5	91.5	**82.7**
Path	0.6	**73.2**	73.9	92.4	**82.1**
		SWI-modified basic Li measure (*from Chapter 3*)			
J&C	0.5	72.5	71.5	97.2	82.4
Path	0.5	71.0	70.2	97.9	81.8
		SWI-modified Mihalcea measure (*from Chapter 3*)			
J&C	0.6	73.7	76.6	86.9	81.4
Path	0.6	73.3	76.1	87.4	81.3
		Islam and Inkpen (2008), Corpus-based			
STS	0.6	72.6	74.7	89.1	81.3
		Mihalcea *et al.* (2006), Corpus-based			
PMI-IR	0.5	69.9	70.2	95.2	81.0
LSA	0.5	68.4	69.7	95.2	80.5
		Mihalcea *et al.* (2006), WordNet-based			
L&C	0.5	69.5	72.4	87.0	79.0
J&C	0.5	69.3	72.2	87.1	79.0
		Ramage *et al.* (2009) Random Graph Walk			
Cosine	0.5	68.7	-	-	78.7
Dice	0.5	70.8	-	-	80.1
JS	0.5	68.8	-	-	80.5
		Baselines			
Vector-based	0.5	65.4	71.6	79.5	75.3
Random	0.5	51.3	68.3	50.0	57.8

As can be seen from Table 5.2, the performance of the SSSE measure exceeds those of the results reported in Mihalcea *et al*. (2006); Ramage *et al*. (2009) and Islam and Inkpen (2008). It can also be seen that it also slightly outperforms the modified measures described in Chapter 3, and far exceeds the baselines. Best performance achieved by a human judge was 83%, which can be considered as an upper bound for an automatic paraphrasing task performed on this dataset.

5.3.2 Textual Entailment Recognition

Table 5.3 compares the performance of the SSSE measure with various other measures on the RTE2 and RTE3 datasets. Note that two sets of results are reported in Ramage *et al*., (2009): one in which the Random Graph Walk method is used as a stand–alone measure, and a second in which the graph walk method is incorporated within an existing RTE system (i.e., a system designed specifically to detect entailment) [Chambers *et al*., 2007]. The baseline represents the original performance of this RTE system [Chambers *et al*., 2007].

The performance of the SSSE measure exceeds the performance of all other methods and baselines on the RTE3 dataset. On the RTE2 dataset its performance is slightly below that of the other reported methods, and approximately equal to the baseline.

As noted in Chapter 3, participants in the RTE challenge have used a variety of strategies beyond lexical relatedness (asymmetric relation), and accuracies as high as 75.4% [Dagan *et al*., 2007] and 80% [Hickl and Bensley, 2007] respectively have been reported on the RTE2 and RTE3 datasets.

Table 5.3: Comparison of performance (%) of SSSE measure with other similarity measures and baseline on the RTE2 and RTE3 datasets.

Measure	Threshold	RTE2 Accuracy	RTE3 Accuracy
		SSSE measure	
J&C	0.5	63.8	**68.7**
Path	0.5	62.8	**70.2**
		SWI-modified basic Li measure (*from Chapter 3*)	
J&C	0.5	**64.2**	65.3
Path	0.5	**64.6**	67.0
		SWI-modified Mihalcea measure (*from Chapter 3*)	
J&C	0.5	**64.3**	67.0
Path	0.5	**64.8**	67.8
		Ramage *et al*. [2009] with Random Graph Walk	
Cosine	0.5	57.0	55.7
Jensen-Shannon	0.5	57.5	56.7
		Ramage *et al*. [2009] with existing RTE system	
Cosine	0.5	**64.5**	65.8
Jensen-Shannon	0.5	63.2	65.4
		Baseline	
Existing RTE3	0.5	63.6	65.4

5.3.3 30-Sentence Pairs Dataset

The 30-Sentence Pairs dataset, due to Li *et al*. [2006], and designed to compare the correlation between machine-assigned similarity measures with human-rated similarity, was described in Section 2.5. Table 5.4 compares the individual sentence similarity scores achieved by the SSSE measure (using the J&C word-word measure) with those of several other sentence similarity measures. The human similarity scores are provided as the mean score for each pair, and have been scaled into the range 0.0 to 1.0.

Table 5.4: Human similarity scores along with different sentence similarity measures.

R&G No.	R&G Word Pair in the Sentences	Human Similarity (Mean)	Li *et al.* (2006) Similarity Measure	Islam and Inkpen (2008) Similarity Measure	SSSE measure
1	Cord, Smile	0.01	0.33	0.06	0.08
5	Autograph, Shore	0.01	0.29	0.11	0.08
9	Asylum, Fruit	0.01	0.21	0.07	0.12
13	Boy, Rooster	0.11	0.53	0.16	0.23
17	Coast, Forest	0.13	0.36	0.26	0.26
21	Boy, Sage	0.04	0.51	0.16	0.25
25	Forest, Graveyard	0.07	0.55	0.33	0.30
29	Bird, Woodland	0.01	0.33	0.12	0.18
33	Hill, Woodland	0.15	0.59	0.29	0.32
37	Magician, Oracle	0.13	0.44	0.20	0.25
41	Oracle, Sage	0.28	0.43	0.09	0.26
47	Furnace, Stove	0.35	0.72	0.30	0.30
48	Magician, Wizard	0.36	0.65	0.34	0.31
49	Hill, Mound	0.29	0.74	0.15	0.10
50	Cord, String	0.47	0.68	0.49	0.34
51	Glass, Tumbler	0.14	0.65	0.28	0.29
52	Grin, Smile	0.49	0.49	0.32	0.50
53	Serf, Slave	0.48	0.39	0.44	0.73
54	Journey, Voyage	0.36	0.52	0.41	0.51
55	Autograph, Signature	0.41	0.55	0.19	0.50
56	Coast, Shore	0.59	0.76	0.47	0.73
57	Forest, Woodland	0.63	0.70	0.26	0.40
58	Implement, Tool	0.59	0.75	0.51	0.77
59	Cock, Rooster	0.86	1	0.94	0.92
60	Boy, Lad	0.58	0.66	0.60	0.55
61	Cushion, Graveyard	0.52	0.66	0.29	0.34
62	Cemetery, Graveyard	0.77	0.73	0.51	0.70
63	Automobile, Car	0.56	0.64	0.52	0.73
64	Midday, Noon	0.96	1	0.93	1
65	Gem, Jewel	0.65	0.83	0.65	0.65

Table 5.5 shows the correlation between the various similarity measures and the average human measures. The correlation of 0.877 achieved by the SSSE measure exceeds that of both the Li *et al*. (2006) measure (0.816) and the Islam and Inkpen (2008) measure (0.853). It also exceeds the mean human correlation of 0.825, far exceeds the performance of the worst human participant, but is still some way off from the best human correlation of 0.921.

Table 5.5: Comparison of performance of different sentences similarity measures on 30-Sentence Pairs dataset.

	Worst Human Participant	**Li *et al.* (2006) Measure**	**Mean of all Human Participants**	**Islam and Inkpen (2008) Measure**	**SSSE Measure**	**Best Human Participant**
Correlation	0.594	0.816	0.825	0.853	**0.877**	0.921

5.4 Discussion and Conclusions

The SSSE measure operates by expanding the semantic context in the direction indicated by the sense-assigned meanings of the original words in the sentence, thereby creating an enriched semantic context, and enabling a more accurate estimate of semantic similarity. While Ho *et al*. (2010) used WSD in sentence similarity measurement, and Zhao *et al*. (2006) and Feng *et al*. (2008) used the idea of context expansion, to the author's best knowledge, this research is the first attempt to incorporate *both* WSD and synonym expansion into sentence similarity measurement.

The results from the previous section demonstrate that incorporating WSD and synonym expansion does lead to improvement in sentence similarity measurement. Importantly, this improvement is gained with very little increase in computational cost. Although the chapter has described how these ideas can be incorporated into a measure based on a reduced vector space representation, the ideas can readily be applied to measures such as that of Mihalcea *et al*. (2006), which do not use an explicit vector space representation.

Sentence similarity measures are commonly evaluated *in vitro* on binary classification tasks such as the MSRP or RTE datasets. However, there are a number of problems with evaluating similarity measures in this way. Firstly, performing binary classification requires that a threshold be determined, and this requires a training set. Most researchers who have used these datasets are interested only in unsupervised learning, and usually choose a threshold of 0.5. This choice, however, is *ad hoc*: similarity measures such as those that have been described do not output probabilities, and there is no reason for why a 0.5 threshold should be expected to be optimal. The 30-Sentence Pairs dataset does not involve this problem, since it is used to measure correlations, and thus does not require thresholding. In other words it evaluates *relative*—not *absolute*—similarity scores.

Secondly, performing binary classification does not test the full discriminatory capability of a similarity measure. If a measure achieves good performance on a binary classification task, it does not necessarily follow that the measure will perform well when used within some encompassing task. The next chapter evaluates the SSSE measure *in vivo* in the context of sentence clustering.

Chapter 6

Fuzzy Clustering for Sentence-Level Text

The previous chapter proposed the new sentence similarity measure (SSSE), and evaluated it *in vitro* on several standard sentence similarity evaluation datasets. This chapter has two purposes: (i) to present a novel fuzzy relational clustering algorithm that is well suited for clustering sentence-level text; and (ii) to evaluate the SSSE measure *in vivo* on a number of sentence clustering tasks.

In comparison with hard clustering methods, in which a pattern belongs to a single cluster, fuzzy clustering algorithms allow patterns to belong to all clusters with differing degrees of membership. This is important in domains such as sentence clustering, since a sentence is likely to be related to more than one theme or topic present within a document or set of documents. However, because most sentence similarity measures do not represent sentences in a common metric space, conventional fuzzy clustering approaches based on prototypes or mixtures of Gaussians are generally not applicable to sentence clustering. The new fuzzy clustering algorithm presented in this chapter operates on relational input data; that is, data in the form of a square matrix of pairwise similarities between data objects (e.g., sentences). The algorithm uses a graph representation of the data, and operates in an Expectation-Maximisation framework in which a measure of the graph centrality of an object within the graph is interpreted as a likelihood. Results of applying the algorithm to several sentence clustering tasks in conjunction with the SSSE measure demonstrate that the algorithm is capable of identifying overlapping clusters of semantically related sentences, and that it is therefore of potential use in a variety of text processing tasks including document summarisation, and text mining of a more general nature.

The chapter is structured as follows. The fuzzy clustering algorithm is presented in Section 6.1, and Section 6.2 describes internal and external criteria that can be used to evaluate the clustering performance. Section 6.3 provides empirical results of applying the algorithm to a specially constructed dataset of famous quotations. The performance of the proposed algorithm is compared with that of other clustering algorithms, and performance under SSSE sentence similarity is compared against that resulting from other similarity measures. In order to demonstrate the applicability the algorithm to practical tasks, Section 6.4 reports on the application of the clustering algorithm to a recent new article, and discusses potential use of the algorithm within document summarisation tasks. Section 6.5 contains discussion and conclusions.

6.1 Proposed Algorithm

The algorithm presented in this chapter, which will be referred to as *Fuzzy Relational Eigenvector Centrality-based Clustering Algorithm* (FRECCA), has previously been reported in Skabar and Abdalgader (2011). Inspired by the mixture model approach, the algorithm models the data as a combination of components. However, unlike conventional mixture models, which operate in a Euclidean space and use a likelihood function parameterised by the means and covariances of Gaussian components, the FRECCA algorithm has does not use any explicit density model (e.g., Gaussian) for representing clusters. Instead, a graph representation in which nodes represent objects, and weighted edges represent the similarity between objects (sentences) has been applied. Cluster membership values for each node represent the degree to which the object represented by that node belongs to each of the respective clusters, and mixing coefficients represent the probability of an object having been generated from that component. By applying the PageRank algorithm [Brin and Page, 1998] to each cluster, and interpreting the PageRank score of an object within some cluster as a likelihood, the Expectation-Maximisation (EM) framework [Dempster *et al.*, 1977] can then be used to determine the model parameters (i.e., cluster membership values and mixing coefficients). The result is a fuzzy relational clustering algorithm which is generic in nature, and can be applied to any domain in which the relationship between objects is expressed in terms of pairwise similarities. In presenting the algorithm in full detail, it is useful to briefly review Gaussian mixture models and the EM algorithm.

6.1.1 Mixture Models and the EM Algorithm

FRECCA is motivated by the mixture model approach, in which a density is modeled as a linear combination of C component densities $P(\mathbf{x}|m)$ in the form $\Sigma\pi_m p(\mathbf{x}|m)$, where the π_m are called *mixing coefficients*, and represent the prior probability of data point $\mathbf{x}$ having been generated from component m of the mixture. Assuming that the parameters of each component are represented by a parameter vector $\boldsymbol{\Theta}_m$, the problem is to determine the values of the components of this vector, and this can be achieved using the Expectation-Maximisation (EM) algorithm [Dempster *et al.*, 1977]. Following random initialisation of the parameter vectors $\boldsymbol{\Theta}_m$, $m = 1,\ldots,C$, an Expectation step (E-step), followed by a Maximisation step (M-step), are iterated until convergence. The E-step computes the cluster membership probabilities. For example, assuming spherical Gaussian mixture components, these probabilities are calculated as:

$$P(m \mid \mathbf{x}_i) = \frac{\pi_m \, p(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_m, \hat{\sigma}_m)}{\sum\limits_{k=1..C} \pi_k \, p(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_k, \hat{\sigma}_k)}, \; m = 1,\ldots,C \tag{6.1}$$

where $\hat{\mu}_m$ and $\hat{\sigma}_m$ are the current estimates of the mean and standard deviation respectively of component *m*. The denominator acts as a normalisation factor, ensuring that 0 ≤$P(m \mid \mathbf{x}_i)$≤ 1 and $\sum_{m=1}^{C} P(m \mid \mathbf{x}_i) = 1$. In the M-step, these probabilities are then used to re-estimate the parameters. Again using the spherical Gaussian case,

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{i=1}^{N} P(m \mid \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{N} P(m \mid \mathbf{x}_i)}, \; m = 1,\ldots,C \, , \tag{6.2}$$

$$\hat{\sigma}_m{}^2 = \frac{\sum_{i=1}^{N} P(m \mid \mathbf{x}_i) \left\| \mathbf{x}_i - \hat{\boldsymbol{\mu}}_m \right\|^2}{\sum_{i=1}^{N} P(m \mid \mathbf{x}_i)}, \; m = 1,2,\ldots,C \, , \tag{6.3}$$

$$\pi_m = \frac{1}{N} \sum_{i=1}^{N} P(m \mid \mathbf{x}_i), \; m = 1,\ldots,C \, . \tag{6.4}$$

The $p(\mathbf{x} \mid \mu, \sigma)$ are called ‘likelihoods’, and in the case of Gaussians are simply the value of the Gaussian with mean μ and variance σ^2 evaluated at point **x**.

6.1.2 Fuzzy Relational Clustering

Unlike Gaussian mixture models, which use a likelihood function parameterised by the means and covariances of the mixture components, FRECCA uses the PageRank score of an object within a cluster as a measure of its centrality to that cluster. These PageRank values are then treated as likelihoods. Since there is no parameterised likelihood function as such, the only parameters that need to be determined are the cluster membership values and mixing coefficients. The algorithm uses Expectation Maximisation to optimise these parameters. We assume in the following that the similarities between objects (sentences) are stored in a similarity matrix $S = \{s_{ij}\}$, where s_{ij} is the similarity between objects i and j.

Initialisation: We assume here that cluster membership values are initialised randomly, and normalised such that cluster membership for an object sums to unity over all clusters. Mixing coefficients are initialised such that priors for all clusters are equal. Alternative initialisation schemes are discussed in Section 6.1.4.

Expectation Step: The E-step calculates the PageRank value for each object in each cluster. PageRank values for each cluster are calculated as described in Equation 3.2 (cf., Section 3.1), with the affinity matrix weights w_{ij} obtained by scaling the similarities by their cluster membership values; i.e.,

$$w_{ij}^{m} = s_{ij} \times p_{i}^{m} \times p_{j}^{m} \tag{6.5}$$

where w_{ij}^{m} is the weight between objects i and j in cluster m, s_{ij} is the similarity between objects i and j, and P_{i}^{m} and P_{j}^{m} are the respective membership values of objects i and j to cluster m. The intuition behind this scaling is that an object's entitlement to contribute to the centrality score of some other object depends not only on its similarity to that other object, but also on its degree of membership to the cluster. Likewise, an object's entitlement to receive a contribution depends on its membership to the cluster. Once PageRank scores have been determined, these are treated as likelihoods and used to calculate cluster membership values.

Maximisation Step: Since there is no parameterised likelihood function, the maximisation step involves only the single step of updating the mixing coefficients based on membership values calculated in the Expectation Step.

The pseudocode is presented in Algorithm 1, where w_{ij}^m, s_{ij}, P_i^m and P_j^m are defined as above, π_m is the mixing coefficient for cluster *m*, PR_i^m is the PageRank score of object *i* in cluster *m*, and l_i^m is the likelihood of object *i* in cluster *m*.

Algorithm 1. The *FRECCA* algorithm.

```
Input: Pairwise similarity values $S=\{s_{ij} \mid i=1,...,N, j=1,...,N\}$ where $s_{ij}$ is the similarity
       between sentences i and j.
       Number of clusters, C.

Output: Cluster membership values $\{p_i^m \mid i=1,...,N, m=1,...,C\}$

// INITIALISATION
// initialise and normalise membership values
for i = 1 to N
    for m = 1 to C
        $P_i^m$ = rnd                // random number on [0, 1]
    end for
    for m = 1 to C
        $p_i^m = p_i^m \Big/ \sum_{j=1}^{C} p_i^j$    // normalise
    end for
end for
for m = 1 to C
    $\pi_m = 1/C$                // equal priors
end for
repeat until convergence
    // EXPECTATION STEP
    for $m = 1$ to $C$
        // create weighted affinity matrix for cluster m
        for $i = 1$ to $N$
            for $j = 1$ to $N$
                $w_{ij}^{m} = s_{ij} \times p_i^m \times p_j^m$
            end for
        end for
        // calculate PageRank scores for cluster m
        repeat until convergence
            $PR_i^m = (1-d) + d \times \sum_{j=1}^{N} w_{ji}^{m} \left( PR_j^m \Big/ \sum_{k=1}^{N} w_{jk}^{m} \right)$
        end repeat
        // assign PageRank scores to likelihoods
        $l_i^m = PR_i^m$
    end for
    // calculate new cluster membership values
    for $i = 1$ to $N$
        for $m = 1$ to $C$
            $p_i^m = \left( \pi_m \times l_i^m \right) \Big/ \sum_{j=1}^{C} \left( \pi_j \times l_i^j \right)$
        end for
    end for
    // MAXIMISATION STEP
    // Update mixing coefficients
    for $m = 1$ to $C$
        $\pi_m = \frac{1}{N} \sum_{i=1}^{N} p_i^m$
    end for
end repeat
```

6.1.3 Sentence Similarity Measure and Thresholding Values

In the case of sentence clustering, the similarity values s_{ij} for the affinity matrix S can be determined using an appropriate sentence similarity measure such as the SSSE measure described in the previous chapter. In most cases the similarity values will be non-zero, leading to a heavily connected graph. Also, many of the similarity values will be very small, arising from incidental similarities between words in sentences which are in fact not semantically related. In practice, we have found that the clustering performance of the algorithm can be improved by thresholding these similarity values such that all values below the threshold are converted to zero. All sentence clustering results reported in this chapter are based on thresholding similarity values such that 50% of the values in the affinity matrix are zero (i.e., other threshold values were investigated e.g., between 30% and 70%, but it was found that performance was not highly sensitive to this).

6.1.4 Convergence and Complexity

With regard to space complexity, the FRECCA algorithm is no more expensive than either the Spectral Clustering [Luxburg, 2007] or *k*-Means [MacQueen, 1967] families of algorithms, since all require the storage of the same, potentially large, similarity matrix. However, the time complexity of FRECCA far exceeds that of both Spectral Clustering and *k*-Means. Whereas Spectral Clustering performs a single eigenvalue decomposition (equivalent to applying a single instance of PageRank), FRECCA calls PageRank on each cluster during each Expectation step. Moreover, membership values must be normalised following the computation of likelihoods. This will perturb the within-cluster PageRank values, and means that the outer repeat loop commencing at Line 14 may be slow to converge, particularly if cluster membership values are initialised randomly. Gaussian mixture models do not have this problem because they represent clusters using means and covariances, which are more stable under changes in cluster membership.

An alternative to random initialisation is to initialise cluster membership values with values found by first applying a computationally inexpensive hard clustering algorithm such as Spectral Clustering or *k*-Medoids. This will result in each object having an initial membership value of either 0 or 1 to each cluster. In practice we have found this to have a significant effect on the rate of convergence, with convergence typically achieved in 30 to 50 EM cycles—approximately one tenth the number of iterations required when using random initialisation. However, care should be taken that the hard clustering algorithm is not itself highly sensitive to initialisation, and for this reason we prefer Spectral Clustering. Note, however, that initialisation does not affect the final membership values at convergence; that is, on all datasets tested, the algorithm converged to the same solution, irrespective of initialisation.

6.1.5 Duplicate Clusters

The number of initial clusters must be specified as input to the algorithm. If this number is too high, then duplicate clusters (i.e., clusters with identical membership values across all objects) will be found. While it might appear at first sight that duplicate clusters can simply be removed after the algorithm has converged, and membership subsequently re-normalised to sum to one, this is not possible because of the coupling between membership values and PageRank values. That is, we cannot assume that the current PageRank values will be correct under a re-normalisation of membership values. The solution is to perform a check for duplicate clusters at the completion of each Maximisation step. If duplicate clusters are found, membership values are re-normalised, and the algorithm is allowed to proceed until a stage at which convergence has been achieved and no duplicate clusters exist.

6.1.6 Effect of Damping Factor, *d*

The damping factor *d* that appears in the PageRank calculation affects the *fuzziness* of the clustering, but generally does not affect the number of clusters, provided that the value is above approximately 0.8. This was observed through conducting a number of trials. In general, the higher the value of *d*, the harder is the clustering, with cluster membership values being close to either zero or one. The value of 0.85 has been used in all of the experiments described in this chapter.

6.1.7 Alternative Eigenvector Centrality Measures

PageRank centrality can be viewed as a special case of eigenvector centrality [Brandes and Erlebach, 2005]. Suppose that a graph is fully connected, in which case it is no longer necessary to reserve some probability of jumping to a random node. In this case, *d* can be set to 1, and the PageRank calculation reduces to solving the eigenvector equation $SC = \lambda C$, where $S = \{s_{ij}\}$ is the affinity matrix containing row-normalised pairwise similarities, λ is the largest eigenvector of this equation, and C is the eigenvector corresponding to this eigenvalue (referred to as the 'dominant eigenvector'), the i^{th} value of which provides a measure of the relative centrality of node i within the graph. By the Perron-Frobenius theorem [Grimmett and Stirzaker, 2001], the dominant eigenvector will always have all non-negative components, thus satisfying the requirement that node centrality scores be non-negative, and in principle any eigenvalue algorithm can be used to find this dominant eigenvector.

PageRank belongs to the family of power iteration methods, which begin with a random vector C_0, and iterate the step $C_{k+1} = SC_k$ until convergence, at which point C will be the dominant eigenvector. Algorithms based on matrix decomposition techniques can also be applied, and avoid the need for iteration. However these may fail due to bad scaling unless the similarity matrix is appropriately normalised. A common choice in the Spectral Clustering literature is to use the graph Laplacian, defined as $Sp = D^{-1/2} S D^{-1/2}$, where diagonal elements d_{ii} of D are equal to the sum of weights in row i of S, and non-diagonal elements are zero. We emphasise, however, that in all sentence clustering experiments performed in this chapter the resulting graphs are sparsely connected, and consequently all results are based on PageRank centrality.

6.1.8 Hard Clustering

The algorithm outputs cluster membership values P_i^m, which represent the degree of membership of object i to cluster m. If hard clustering is required, this can be trivially achieved by assigning a sentence to the cluster m for which membership is highest; i.e., $\arg\max_{m \in C} \{P_i^m\}$.

6.2 Cluster Evaluation Criteria

Cluster evaluation may be either *supervised*, in which case external information (usually known class labels associated with the instances) is used to measure the goodness of the clustering; or *unsupervised*, in which case no external information is used. In the following $L = \{w_1, w_2, \ldots\}$ is the set of clusters, $C = \{c_1, c_2, \ldots\}$ is the set of classes (for supervised evaluation), and N is the number of objects (sentences).

6.2.1 Partition Entropy Coefficient

Many unsupervised evaluation measures have been defined, but most are only applicable to clusters represented using prototypes. Two exceptions are the Partition Coefficient (PC) [Bezdek, 1974] and the closely related Partition Entropy Coefficient (PE) [Bezdek, 1975], the latter of which is defined as

$$PE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{|L|}\left(u_{ij}\log_a u_{ij}\right) \tag{6.6}$$

where u_{ij} is the membership of instance i to cluster j. The value of this index ranges from 0 to $\log_a|\mathrm{L}|$. The closer the value is to 0, the crisper the clustering is. The highest value is obtained when all of the u_{ij}s are equal. The remainder of the criteria that we will describe are all supervised.

6.2.2 Purity and Entropy

Two widely used external clustering evaluation criteria are purity and entropy [Manning *et al.*, 2008]. The purity of a cluster is defined as the fraction of the cluster size that the largest class of objects assigned to that cluster represents; thus, the purity of cluster *j* is

$$P_j = \frac{1}{|w_j|}\max_i\left(|w_j \cap c_i|\right) \tag{6.7}$$

Overall purity is just the weighted average of the individual cluster purities:

$$Overall\,Purity = \frac{1}{N}\sum_{j=1}^{|L|}\left(|w_j| \times P_j\right) \tag{6.8}$$

The entropy of a cluster *j* is a measure of how mixed the objects within the cluster are, and is defined as

$$E_j = -\frac{1}{\log|C|}\sum_{i=1}^{|C|}\frac{|w_j \cap c_i|}{|w_j|}\log\frac{|w_j \cap c_i|}{|w_j|} \tag{6.9}$$

Overall entropy is the weighted average of the individual cluster entropies:

$$Overall\,Entropy = \frac{1}{N}\sum_{j=1}^{|L|}\left(|w_j| \times E_j\right) \tag{6.10}$$

Good clustering is thus characterised by a high purity and low entropy.

Because entropy and purity measure how the classes of objects are distributed within each cluster, they measure *homogeneity*; i.e., the extent to which clusters contain only objects from a single class. However, we are also interested in *completeness*; i.e., the extent to which all objects from a single class are assigned to a single cluster. While high purity and low entropy are generally easy to achieve when the number of clusters is large, this will result in low completeness, and in practice we are usually interested in achieving an acceptable balance between the two.

6.2.3 V-measure

This problem with purity and entropy is overcome by the *V*-measure [Rosenberg and Hirschberg, 2007], also known as the Normalised Mutual Information (NMI) [Manning *et al*., 2008], which is defined as the harmonic mean of homogeneity (*h*) and completeness (c); i.e.,

$$V = hc \,/\, (h + c) \tag{6.11}$$

where *h* and *c* are defined as

$$h = 1 - \frac{H(C \mid L)}{H(C)} \quad \text{and} \quad c = 1 - \frac{H(L \mid C)}{H(L)}$$

where

$$H(C) = -\sum_{i=1}^{|C|} \frac{|c_i|}{N} \log \frac{|c_i|}{N}, \quad H(L) = -\sum_{j=1}^{|L|} \frac{|w_j|}{N} \log \frac{|w_j|}{N}$$

$$H(C \mid L) = -\sum_{j=1}^{|L|} \sum_{i=1}^{|C|} \frac{|w_j \cap c_i|}{N} \log \frac{|w_j \cap c_i|}{|w_j|}, \text{ and}$$

$$H(L \mid C) = -\sum_{i=1}^{|C|} \sum_{j=1}^{|L|} \frac{|w_j \cap c_i|}{N} \log \frac{|w_j \cap c_i|}{|c_i|}.$$

Because it takes into account both homogeneity and completeness, *V*-measure is more reliable than purity or entropy when comparing clusterings with different numbers of clusters.

6.2.4 Rand Index and F-measure

Unlike purity, entropy and V-measure, which are based on statistics, Rand Index and F-measure are based on a combinatorial approach which considers each possible pair of objects. Each pair can fall into one of four groups: if both objects belong to the same class and same cluster then the pair is a true positive (TP); if objects belong to the same cluster but different classes the pair is a false positive (FP); if objects belong to the same class but different clusters the pair is a false negative (FN); otherwise the objects must belong to different classes and different clusters, in which case the pair is a true negative (TN). The Rand index [Rand, 1971] is simply the accuracy; i.e., $RI = (TP + FP)/(TP + FP + FN + TN)$. The *F*-measure is another measure commonly used in the IR literature, and is defined as the harmonic mean of precision and recall; i.e., *F*-measure = $2PR/(P + R)$, where $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$.

6.3 Clustering Famous Quotations

This section reports on the application of the algorithm to two specially constructed datasets of famous quotations. The performance of the proposed algorithm is compared with that of other clustering algorithms, and performance under SSSE sentence similarity is compared against that resulting from other sentence similarity measures.

6.3.1 Famous Quotations Datasets

Famous quotations provide a rich and challenging context for evaluating sentence clustering because they often contain a lot of semantic information (i.e., wisdom packed into a small message), and are often couched in a poetic use of language. Two quotations datasets have been constructed: the 50-Quotes dataset, and the 211-Quotes dataset. The first dataset contains 50 quotes from 5 different classes (knowledge, marriage, nature, peace, food). The quotations are equally distributed among classes; i.e., ten quotes from each class. The second dataset contains 211 quotes from 15 different classes (politics, music, education, success, work, forgiveness, experience, health, law, spirituality, marriage, food, intelligence, peace, money). In this case the quotes are not equally distributed amongst classes. Quotes in the 211-Quotes dataset were deliberately selected to display a lower degree of word co-occurrence than those in the 50-Quotes dataset, and can thus be expected to be more difficult to cluster.

Extracts from the 50-Quotes and 211-Quotes datasets are shown in Tables 6.1 and 6.2 respectively. Full datasets are provided in Appendix B. The quotes are taken from the Famous Quotes and Authors website (http://www.famousquotesandauthors.com/, accessed 26 March 2011).

Table 6.1: Extract from the 50-Quotes dataset (two quotations from each of five classes).

Knowledge

1. Our knowledge can only be finite, while our ignorance must necessarily be infinite.
2. Everybody gets so much common information all day long that they lose their commonsense.

…

Marriage

11. A husband is what is left of a lover, after the nerve has been extracted.
12. Marriage has many pains, but celibacy has no pleasures.

…

Nature

21. I have called this principle, by which each slight variation, if useful, is preserved, by the term natural selection.
22. Nature is reckless of the individual; when she has points to carry, she carries them.

…

Peace

31. There is no such thing as inner peace, there is only nervousness and death.
32. Once you hear the details of victory, it is hard to distinguish it from a defeat.

…

Food

41. Food is an important part of a balanced diet.
42. To eat well in England you should have breakfast three times a day.

…

Table 6.2: Extract from the 211-Quotes dataset (all quotations from class Success).

43. The world belongs to the enthusiast who keeps cool.
44. Men at some time are masters of their fates.
45. The secret of success is constancy to purpose.
46. Survival is triumph enough.
47. The secret of all victory lies in the organization of the non obvious.
48. The conditions of conquest are always easy. We have but to toil awhile, endure awhile, believe always, and never turn back.
49. The very first step towards success in any occupation is to become interested in it.
50. Four steps to achievement: plan purposefully, prepare prayerfully, proceed positively, pursue persistently.
51. Always aim for achievement, and forget about success.
52. The way to rise is to obey and please.

…

6.3.2 Clustering the 50-Quotes Dataset

Tables 6.3, 6.4 and 6.5 show the results of applying the FRECCA, ARCA (cf., Section 2.4.2) and Spectral Clustering (cf., Section 2.4.2) algorithms respectively to the 50-Quotes dataset and evaluating using the external measures described above. In order to compare the effect of the sentence similarity measures, the first section of each table shows performance with the use of the SSSE measure and the second shows performance with the use of basic Li measure. The spectral clustering algorithm used is that due to Ng *et al*. (2001). Note that FRECCA and ARCA identify fuzzy clusters, and external evaluation requires that the fuzzy cluster membership values first be converted to crisp (i.e., 0/1) values.

FRECCA requires that an initial number of clusters is specified. This number was varied from 3 to 15. Interestingly, only 3 unique clusterings were found in the case of using the SSSE measure and only 6 unique clusterings were found in case of using the basic Li measure, each containing a different number of clusters, which ranged from 3 to 5 and 3 to 8 respectively. We emphasise that the tabulated results for FRECCA are not averages—these were the *only* clusterings found.

The ARCA algorithm is based on Fuzzy *c*-Means [Dunn, 1973], and therefore requires selecting a value for the weighting exponent r ($r \geq 1$) that controls the fuzziness of the resulting clusters. When $r = 1$, the algorithm reduces to the basic *k*-Means algorithm [MacQueen, 1967]; as r is increased, so too is the degree of fuzziness. The best external clustering results were obtained for an r value of 1.125.

In the case of Spectral Clustering there are no parameters to specify other than the number of clusters. Values in the tables are averaged over 3 trials.

Since the five performance measures are not always consistent as to which algorithm achieves best performance for a given number of clusters, we indicate in boldface the value corresponding to the best value for that measure; i.e., the maximum column value in the case of Purity, *V*-measure, Rand Index and *F*-measure, and the minimum column value in the case of Entropy.

In each of the three tables, it can clearly be seen that use of the SSSE measure consistently leads to better clustering performance over that of the basic Li measure. That is, the SSSE measure leads to better performance across all three algorithms.

Comparing the first sections of Tables 6.3, 6.4 and 6.5 (i.e., performance % of the three algorithms using the SSSE measure) shows that the FRECCA algorithm outperforms the ARCA and Spectral Clustering algorithms. The FRECCA algorithm also achieves superior results to that of the other algorithms when using the basic Li measure, as can be seen by comparing the second sections of the three tables.

Best performance in terms of overall purity, entropy, V-measure, rand-index and F-measures (88.0%, 24.8%, 75.5%, 91.0% and 75.9% respectively), was achieved using FRECCA with SSSE measure. Interestingly, note that this best performance occurs when the number of clusters is five, which happens to be the actual number of clusters in the dataset.

Table 6.3: Clustering evaluation on 50-Quotes dataset using FRECCA.

N_clust	Purity	Entropy	*V*-measure	Rand Index	*F*-measure
			SSSE Measure		
3	0.480	0.727	0.324	0.699	0.404
4	0.660	0.488	0.553	0.813	0.562
5	**0.880**	**0.248**	**0.755**	**0.910**	**0.759**
			Basic Li measure		
3	0.500	0.723	0.331	0.704	0.419
4	0.640	0.543	0.496	0.787	0.510
5	0.800	0.352	0.652	0.864	0.636
6	0.800	0.324	0.646	0.862	0.601
7	0.680	0.437	0.513	0.807	0.423
8	0.740	0.364	0.559	0.827	0.421

Table 6.4: Clustering evaluation on 50-Quotes dataset using ARCA.

N_clust	**Purity**	**Entropy**	***V*-measure**	**Rand Index**	***F*-measure**
SSSE Measure					
3	0.540	0.566	0.530	0.745	0.537
4	0.660	0.485	0.576	0.784	0.552
5	0.700	0.450	0.564	0.809	0.529
6	0.800	0.341	0.629	0.860	**0.595**
7	0.800	0.290	**0.646**	**0.862**	0.564
8	**0.820**	**0.276**	0.637	0.861	0.538
Basic Li measure					
3	0.440	0.764	0.290	0.638	0.357
4	0.540	0.648	0.392	0.708	0.387
5	0.620	0.512	0.497	0.788	0.466
6	0.620	0.467	0.507	0.805	0.430
7	0.740	0.386	0.568	0.834	0.496
8	0.720	0.421	0.507	0.822	0.398

Table 6.5: Clustering evaluation on 50-Quotes dataset using Spectral Clustering.

N_clust	**Purity**	**Entropy**	***V*-measure**	**Rand Index**	***F*-measure**
SSSE Measure					
3	0.500	0.641	0.463	0.620	0.430
4	0.700	0.401	0.666	0.821	0.620
5	0.740	0.394	0.616	0.827	0.559
6	0.760	0.309	0.667	0.851	0.582
7	**0.840**	0.267	**0.678**	**0.880**	**0.638**
8	0.840	**0.242**	0.670	0.872	0.582
Basic Li measure					
3	0.471	0.726	0.358	0.577	0.382
4	0.641	0.472	0.583	0.795	0.554
5	0.652	0.508	0.508	0.785	0.477
6	0.690	0.475	0.508	0.800	0.444
7	0.699	0.429	0.530	0.809	0.431
8	0.701	0.415	0.521	0.815	0.406

A more intuitive appreciation of the clustering performance can be gained by examining the quotations assigned to the various clusters. This is shown for the 5-clustering in Table 6.6, where the labels in parentheses indicate the majority class (i.e., the class of quotations most frequent in that cluster), and numbers in boldface represent quotations belonging to that class. The first section of the table shows the results of clustering using FRECCA with the SSSE measure; the second shows the results of clustering using FRECCA with the basic Li measure. In the first section of the table, Clusters 2 and 4 are completely homogenous, since they contain quotes from only a single class. Each of the other clusters contains one or two quotes not belonging to the class of the majority of quotes in the cluster. In the second section of the table, there are no perfectly homogeneous clusters, and in one case (Cluster 5) there are four quotes not belonging to the majority class. In regard to completeness, there is little difference between the two clusterings.

The better performance achieved using the SSSE measure is most likely due its ability to capture more semantic information than the basic Li measure. To illustrate, consider Quotation 32: "*When fire and water are at war it is the fire that loses.*", which belongs to the actual class *Peace*. When clustered using the basic Li measure, this quote is clustered with quotes belonging predominantly to class *Food*, probably due to the presence of the word 'water', which might be considered a type of food, and also possibly due to the presence of the word 'fire' (used for cooking food). However, when clustered using SSSE similarity, the quote is clustered into the same cluster as almost all other quotes belonging to class *Peace*, most likely due to the presence of the word 'war'. The most likely explanation for this is that the SSSE measure, because it uses an expanded semantic context, is better able to make a stronger connection between 'war' and war-related words appearing in other quotes belonging to Class *Peace*.

Table 6.6: Hard cluster assignment for FRECCA using the SSSE and the basic Li similarity measures.

Cluster	Sentences belonging to cluster.
SSSE Measure	
1	**1, 2, 3, 4, 5, 6, 7, 8, 9,** 13, 21 ('Knowledge')
2	**14, 15, 16, 17, 18, 19, 20** ('Marriage')
3	10, 12, **22, 23, 24, 25, 26, 27, 28, 29, 30** ('Nature')
4	**32, 33, 34, 35, 36, 37, 38, 39, 40** ('Peace')
5	11, 31, **41, 42,43, 44, 45, 46, 47, 48, 49, 50** ('Food')
Basic Li measure	
1	**1, 2, 3, 4, 5, 6, 7, 8,** 11, 12 ('Knowledge')
2	9, **15, 16, 17, 18, 19, 20** ('Marriage')
3	10, **22, 23, 24, 25, 26, 27, 28, 29, 30** ('Nature')
4	21, **33, 34, 35, 36, 37, 38, 39, 40,** 50 ('Peace')
5	13, 14, 31, 32, **41, 42,43, 44, 45, 46, 47, 48, 49,** ('Food')

Fuzzy Clustering

The above results for FRECCA are based on its performance in hard clustering mode; that is, cluster membership is determined by assigning a quotation to the cluster for which its membership is highest. However, an advantage of FRECCA is that it assigns soft membership values which can be interpreted as a measure of the degree to which an object belongs to each of the clusters. In some cases, a quote belongs almost exclusively to a single cluster. For example, Quotation 38 "*To be prepared for war is one of the most effectual means of preserving peace*" (see Table B.1) belongs to Cluster 4 with a membership of 0.697, its membership to each of the other clusters being in the vicinity of 0.07. In contrast, Quotation 47 "*To a man with an empty stomach, food is god*" (see Table B.1) has a membership of 0.317 to Cluster 5, which contains all ten quotes belonging to category *Food*, but it also has a relatively high membership of 0.224 and 0.210 to Clusters 2 and 3 respectively, which contain almost exclusively quotes from the *Marriage* and *Nature* categories respectively, some of which have a creationist theme. The ability to assign such a confidence measure is clearly a useful property for clustering sentences in natural language, since most sentences can be considered to belong to a range of topics. The fuzzy membership will be further explored in Section 6.4.

It is informative to compare the fuzzy clusterings obtained by the FRECCA and ARCA algorithms using the Partition Entropy Coefficient (PE) defined in Section 6.2.1. Whereas FRECCA achieves a PE value of 1.29, indicating a high degree of overlap between clusters (the maximum possible PE value for this dataset is 1.61), ARCA achieves a PE value of only 0.015, indicating an almost crisp clustering. While the degree of fuzziness for ARCA can be increased simply by increasing the value of *r*, this was found experimentally to result in a sharp decline in performance as measured by the external criteria.

6.3.3 Clustering the 211-Quotes Dataset

Tables 6.7, 6.8 and 6.9 show the results of applying the FRECCA, ARCA and Spectral Clustering algorithms respectively to the 211-Quotes dataset (see Table B.2). We follow the same evaluation setting as per the 50-Quotes dataset, with the exception that the initial number of clusters was varied from 12 to 18.

The values of the performance measures clearly indicate that the 211-Quotes dataset is a much more challenging dataset of sentences to cluster that is the 50-Quotes dataset. Nevertheless, the same conclusions can be drawn as was the case for the 50-Quotes dataset. That is, all three algorithms achieve better performance with the SSSE measure, and the FRECCA algorithm performs better than other two algorithms, irrespective of which similarity measure is used.

Table 6.7: Clustering evaluation on 211-Quotes dataset using FRECCA.

N_clust	Purity	Entropy	*V*-measure	Rand Index	*F*-measure
			SSSE Measure		
12	0.319	0.683	0.327	0.874	0.173
13	0.376	0.645	0.362	0.884	0.201
14	0.376	0.643	0.358	0.889	0.194
15	0.395	0.622	0.374	0.894	**0.200**
16	0.371	0.621	0.370	0.896	0.182
17	0.371	0.619	0.371	0.895	0.178
18	**0.414**	**0.577**	**0.404**	**0.903**	0.181
			Basic Li measure		
12	0.281	0.731	0.275	0.869	0.127
13	0.290	0.718	0.284	0.877	0.128
14	0.300	0.687	0.316	0.876	0.142
15	0.319	0.662	0.336	0.883	0.151
16	0.314	0.664	0.332	0.883	0.143
17	0.347	0.628	0.361	0.893	0.153
18	0.352	0.641	0.345	0.894	0.136

Table 6.8: Clustering evaluation on 211-Quotes dataset using ARCA.

N_clust	Purity	Entropy	*V*-measure	Rand Index	*F*-measure
			SSSE Measure		
12	0.242	0.790	0.219	0.852	0.105
13	0.242	0.769	0.237	0.859	0.107
14	0.271	0.728	0.276	0.870	0.116
15	0.290	0.706	0.295	0.874	0.122
16	0.295	0.702	0.295	0.878	0.115
17	0.304	0.694	0.302	0.878	0.116
18	**0.309**	**0.678**	**0.312**	**0.886**	**0.118**
			Basic Li measure		
12	0.252	0.766	0.241	0.856	0.110
13	0.252	0.751	0.252	0.867	0.104
14	0.252	0.731	0.271	0.868	0.107
15	0.271	0.732	0.265	0.875	0.101
16	0.276	0.710	0.283	0.881	0.101

17	0.266	0.706	0.285	0.883	0.095
18	0.290	0.691	0.298	0.884	0.099

Table 6.9: Clustering evaluation on 211-Quotes dataset using Spectral Clustering.

N_clust	Purity	Entropy	*V*-measure	Rand Index	*F*-measure
SSSE Measure					
12	0.271	0.776	0.251	0.802	0.142
13	0.304	0.735	0.279	0.849	0.142
14	0.295	0.746	0.275	0.823	0.140
15	0.300	0.742	0.288	0.790	0.145
16	**0.342**	0.693	0.325	0.840	**0.158**
17	0.328	**0.670**	**0.340**	0.855	0.148
18	0.338	0.689	0.318	**0.861**	0.146
Basic Li measure					
12	0.271	0.767	0.267	0.774	0.136
13	0.285	0.762	0.276	0.747	0.139
14	0.290	0.753	0.281	0.780	0.143
15	0.333	0.710	0.321	0.805	0.156
16	0.333	0.705	0.322	0.808	0.149
17	0.361	0.694	0.334	0.807	0.152
18	0.352	0.694	0.325	0.832	0.146

It is interesting to note the relatively poor performance of the ARCA algorithm on this dataset. This is almost certainly due to curse of dimensionality problems arising from the fact that in this case ARCA must perform performing fuzzy means in a space of 211 dimensions. Note that the best performance values (represented in boldface) are achieved when the number of clusters is increased. Unlike the 50-Quotes dataset, in which there is a relatively distinct peak in values of measures such as the V-measure, the 211-Quotes dataset is much more difficult to cluster; thus we do not see a distinct peak, but rather, a tendency for measures to rise slowly as the number of clusters in increased.

Taking all measures into account, by far the best overall performance of the three algorithms is achieved by FRECCA in conjunction with the SSSE measure. In this experiment, however, we knew *a priori* what the actual number of classes (clusters) was. For example, 5-classes in case of 50-Quotes dataset and 15-classes in case of 211-Quotes dataset. In general, we would not have this information, and would hope that the algorithm could automatically determine an appropriate number of clusters. Even when run with a high initial number of clusters, FRECCA was able to converge to a solution containing not more than five clusters (e.g., in case of 50-Quotes), and from the tables it can be seen that the evaluation of these clusterings is better than that for the other clustering algorithms.

6.4 Clustering Sentences from a News Article

The famous quotations dataset was constructed in order to evaluate performance of the algorithm using standard external cluster quality criteria. To demonstrate how the algorithm may be of more general use in activities related to text mining, we now apply the FRECCA to clustering sentences from a recent news article.

Table 6.10 shows the sentences from an article about President Barak Obama's presidency, chosen because it was topical at the time of conducting the study, and that it is typical in terms of length and breadth of content to the type of texts to which text processing activities such as text summarisation are commonly applied. Sentences in boldface are those that the clustering algorithm identified as being central to each of the identified clusters (see later).

Running FRECCA on this dataset results in five clusters, shown graphically in Figure 6.1 to 6.5. Nodes on the graphs represent sentences, and are positioned such that the distance between nodes is in inverse proportion to their similarity. For clarity, edges between nodes for which the similarity measure is 0.25 or greater have only been shown. Thus, nodes placed towards the outside of the graph tend to be semantically related to only a small number of other nodes, whereas nodes placed more toward the centre of the graph are related to many other nodes.

Table 6.10: News Article dataset.

1. **Prodding Republicans, President Barack Obama on Tuesday championed nuclear energy expansion as the latest way that feuding parties can move beyond the "broken politics" of Washington that have imperiled his agenda and soured voters.**
2. His call came as he dispatched Vice President Joe Biden and Cabinet secretaries nationwide to tout the economic stimulus plan against Republican criticism, reflecting that until bipartisanship comes, the White House will remain aggressive in selling its own case to the public.
3. Since a January special election in Massachusetts, when Democrats lost the 60th vote they need in the Senate to overcome Republican delays on legislation, Obama has recalibrated his strategy to advance his agenda.
4. His plan includes reaching out to Republicans on tax breaks, on health care and on energy, but also putting them on the spot for any refusal to help.
5. With a host of new goals — rebuilding public confidence, keeping Obama in charge of the debate, halting deep Democratic losses in this year's elections — the White House is now infusing its communications strategy with more of the discipline that it famously used in Obama's presidential campaign.
6. The president cast his push for more nuclear energy as both economically vital and politically attractive to the opposition party.
7. He announced more than $8 billion in loan guarantees to build the first nuclear power plant in nearly three decades, part of a nuclear initiative that could draw essential backing from Republicans.
8. At the same time, he asked Republicans to get behind a comprehensive energy bill that expands clean energy sources, assigns a cost to the polluting emissions of fossil fuels so that nuclear fuel becomes more affordable, and gives both parties a rare chance to claim common ground.
9. **"The fact is, changing the ways we produce and use energy requires us to think anew.**
10. It requires us to act anew," Obama said during a stop a job training center outside Washington.
11. **"And it demands of us a willingness to extend our hand across some of the old divides, to act in good faith, and to move beyond the broken politics of the past.**
12. That mission, however, remains in doubt.
13. **A White House built on the long view also has gotten sharper about responding to daily criticisms from emboldened Republicans.**
14. This week, senior administration officials are scheduled to visit 35 communities to counter Republican claims that the massive, deficit-spending economic stimulus program has failed.
15. In Saginaw, Mich, on Tuesday, Biden insisted the stimulus is working even as he acknowledged "it's gonna take us a while to get us out of this ditch."
16. Michigan's unemployment rate is among the highest in the country.
17. The chronic joblessness there and elsewhere is driving an anti-incumbency fever, even as the economy by most other measures appears to be rebounding.
18. **Democrats, as members of the party in power, are most likely to feel that anti-incumbency heat at the polls in November when House and Senate seats are on the ballot.**
19. Obama will head west later this week to raise money for two vulnerable Democrats who face the voters this year, Senator Michael Bennet of Colorado and Senator Harry Reid of Nevada, the majority leader.
20. On Tuesday, an Obama ally and moderate Democrat, two-term Senator Evan Bayh of Indiana, said the frustrations of gridlock drove his decision not to run for re-election.
21. "There's just too much brain-dead partisanship," Bayh said in a nationally broadcast interview.
22. Obama is working to change that system while, for now, he is required to work within it.
23. He made his pitch for nuclear energy by saying nothing less than the economy, the security of the United States and the planet's future were at stake.
24. "We can't continue to be mired in the same old stale debates between left and right, between environmentalists and entrepreneurs," the president said.
25. Obama aides say there's no formal reevaluation of the administration's communications strategy as the president embarks on his second year in office.
26. But the White House is taking an approach that is at once more aggressive and more streamlined.
27. It includes more direct, rapid response to criticism; more events at which the president speaks directly to the public without the filter of the media; and more carefully choreographed interactions with the press.
28. The intended narrative is one in which Obama hears people's frustrations and is working directly to end them.
29. There's little doubt the public is angry.
30. A CBS News/New York Times poll in early February found 81 percent saying it's time to elect new people to Congress.
31. That affects Obama, who is not up for re-election until 2012 but needs allies and votes on Capitol Hill to usher in the domestic change he has promised.

The size of nodes in Figure 6.1 to 6.5 represents the node's membership value to the cluster (i.e., the degree of membership to the cluster). Nodes positioned more towards the center of a graph tend to be smaller because they belong to many clusters. (Recall that cluster membership values of a sentence sum to one over all clusters.) Nodes marked by open circles are those with the highest PageRank score for that cluster (i.e., cluster centroids). We refer to these as 'cluster centroids' but note that FRECCA is not a prototype-based algorithm, and hence the centroids should not be considered prototypes. Sentences corresponding to cluster centroids are those shown in boldface in Table 6.10. The clustering is quite evident from the graphs. For example, Cluster 1 (Figure 6.1) contains sentences represented by nodes placed towards the left of the graph; Cluster 2 (Figure 6.2) contains nodes towards the top of the graph, etc.

Manual inspection of the sentences belonging to a cluster can be used as a subjective test to determine whether the cluster centroid reflects the overall sentiment expressed by sentences with high membership to that cluster. For example, sentences belonging to Cluster 2 involve comments about negative public opinion (Sentence 29), high unemployment (Sentences 16 and 17), and the need to elect new people to congress (Sentence 30). These themes are captured appropriately by the cluster centroid, Sentence 18, which involves terms and phrases such as 'anti-incumbency heat', 'polls' and 'seats on the ballot'.

Note that the node with the highest PageRank score in a cluster is generally not the node with the highest membership to that cluster. For example, in Cluster 2 (Figure 6.2), Sentences 16, 23, 29 and 30 (which appear towards the top of the graph) belong almost exclusively to this cluster, whereas the centroid for the cluster is Sentence 18. This is analogous to the Gaussian mixture model case in which an object may belong predominantly to one Gaussian mixture component, while still being very distant from the centroid (i.e., mean) of that (Gaussian) component.

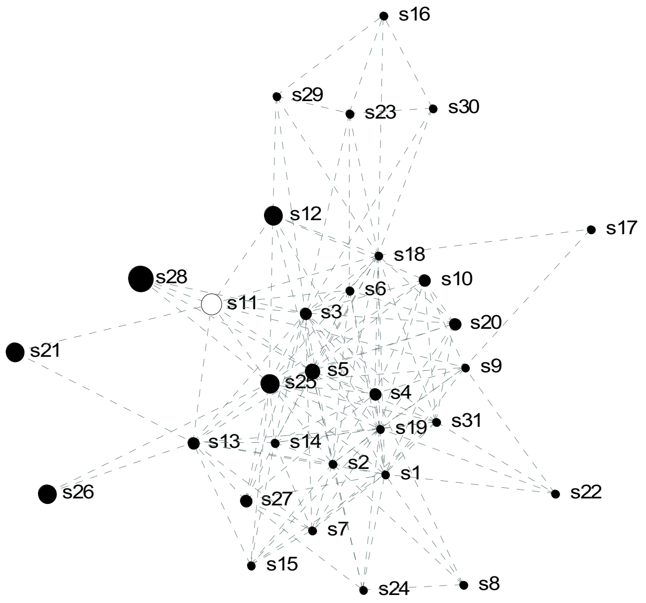

Figure 6.1: Fuzzy membership to Cluster 1 for News Article dataset (Cluster Centroid is s11).

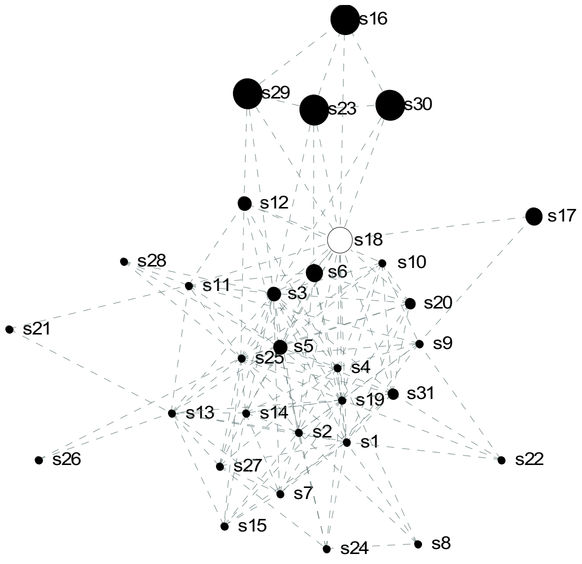

Figure 6.2: Fuzzy membership to Cluster 2 for News Article dataset (Cluster Centroid is s18).

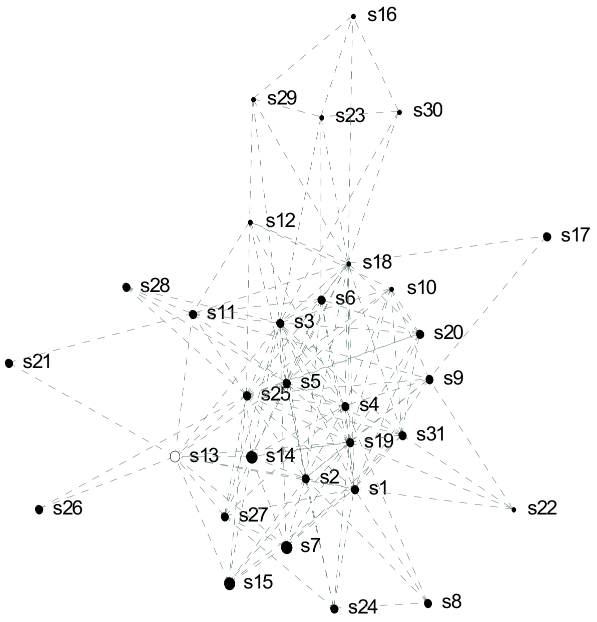

Figure 6.3: Fuzzy membership to Cluster 3 for News Article dataset (Cluster Centroid is s13).

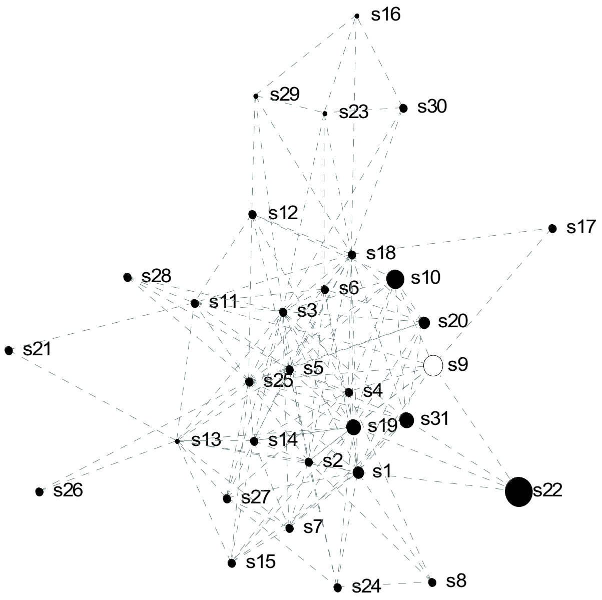

Figure 6.4: Fuzzy membership to Cluster 4 for News Article dataset (Cluster Centroid is s9).

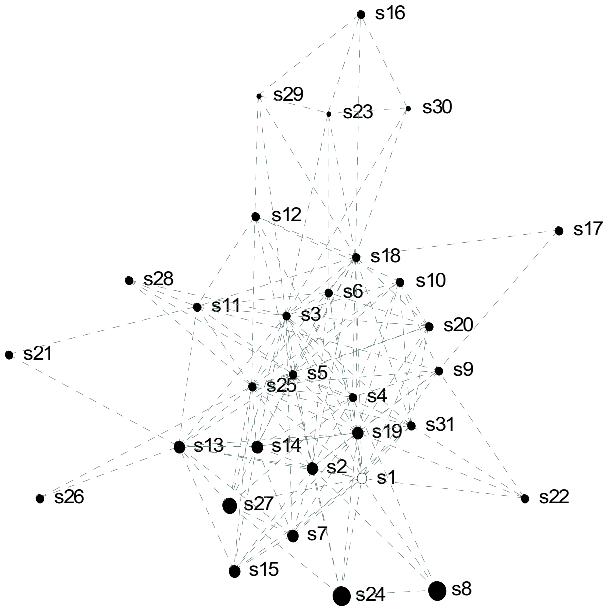

Figure 6.5: Fuzzy membership to Cluster 5 for News Article dataset (Cluster Centroid is s1).

In some cases we may be interested in determining the global importance of a sentence; i.e., how important a sentence is in the context of the article as a whole, irrespective of its membership to individual clusters. This can be achieved by weighting the PageRank scores with cluster mixing coefficients, and summing these weighted scores over all clusters. The reason for weighting by the mixing coefficients is that these coefficients indicate the relative importance of clusters. The resulting value has been referred as the 'global PageRank' (GPR) for the sentence. It is calculated as

$$GPR_i = \sum_{j=1}^{C} \pi_j PR_i^j \tag{6.6}$$

where π_j is the mixing coefficient for cluster *m*, and PR_i^j is the PageRank score of sentence *i* in cluster *j*. The size of nodes in Figure 6.6 represents the node's GPR value. Quite clearly, nodes toward the center of the graph tend to have high global PageRank values, whereas those towards the outer regions have small values. Nodes in open circles are the individual cluster centroids, indicated here for illustrative purposes only.

The GPR values can be used to rank sentences according to their global importance within the document, and we would expect such a ranking to be similar to that produced by algorithms such as LexRank, which apply a single instance of PageRank to the document graph. Indeed, measuring the correlation between GPR scores and centrality scores obtained using a single PageRank instance results in a value of 0.995, indicating near perfect correlation, and verifying that individual cluster centrality scores correctly sum to the expected global score.

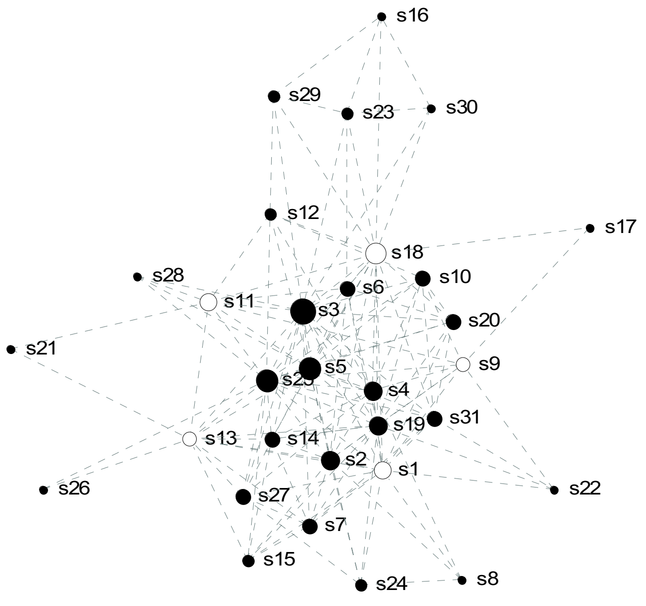

Figure 6.6: Graph shows global PageRank (GPR) scores, indicated by the size of nodes. Cluster centroids are represented as open circles.

Finally, we compare the FRECCA results with those of the ARCA algorithm. ARCA requires that we set the value of the weighting exponent, *r*, which controls the fuzziness of the clustering. Experimenting with a range of values, we found that an *r*-value of 1.25 resulted in a level of clustering deemed to be reasonable, given the fuzzy nature of the domain, and yielded a Partition Entropy Coefficient (PE) value of 1.07, indicating a crisper clustering than that achieved by FRECCA (PE = 1.40). This is consistent with our observations from the quotations dataset, in which ARCA was also found to result in a crisper clustering than FRECCA. Note, however, that the ARCA results are highly sensitive to the value of *r*, with the resulting clusters tending to be either extremely crisp or extremely fuzzy as *r* is decreased or increased. Since ARCA represents clusters using prototypes which do not correspond to particular objects in the dataset, cluster centroids can be considered to be the sentences which are closest (in the Euclidean sense) to the respective prototype. These sentences are identified as Sentences 3, 7, 9, 11 and 16. The position of these sentences on the graphs of Figure 6.1 to 6.6 does not display the even distributedness of centroids identified by FRECCA, and in some cases (e.g., Sentence 16) it would appear that the sentence is a particularly poor representative.

6.4.1 Application to Document Summarisation

Although we have been primarily concerned with sentence clustering as a generic activity, sentence clustering will often be performed within some other text-processing task such as extractive document summarisation, where the objective is to extract a (usually small) subset of sentences to include in a summary.

An obvious way to use the clustering results to produce an extractive summary is to select from each cluster the sentence most central to that cluster. This is trivial in the case of FRECCA, and amounts to simply selecting the centroid from each cluster; i.e., sentences 1, 9, 11, 13 and 18. Note that these sentences tend to be distributed around the perimeter of the denser inner region of the document graph, as can be seen clearly from Figure 6.6. This is intuitively appealing, as we would expect good summary sentences to bear some semantic relationship with each other, while at the same time capturing the breadth of content present in the document. In contrast, methods such as LexRank, which are based on global PageRank scores, would select sentences 3, 5, 25, 19, 2, which correspond to the five largest nodes in Figure 6.6. These sentences tend to be more concentrated toward the center of the graph, suggesting that these sentences may better reinforce the main theme or themes, but possibly not pick up on the some of the minor themes.

Depending on the number of clusters that have been identified, selecting the cluster centroids may result in either too few or too many sentences, and we may wish to either add or delete sentences from this summary. There are various approaches we could take. For example, if we wish to include more sentences, we could select additional sentences from each cluster, but this may result in an overly large summary, with possibly some duplication in content. A better approach would be to supplement the summary with sentences which are important globally within the document, and these sentences can be easily identified by their GPR score. The next five sentences to be added according to this procedure (obviously not adding duplicates) would be Sentences 3, 5, 25, 19, 2, sorted according to their GPR score. It is interesting to note that three of these additional sentences appear very close to the beginning of the article, and intuitively, we would expect the first few sentences of a news article to capture the main content. Indeed, simply selecting the first few sentences in a document is a commonly used benchmark for document summarisation. Should the initial summary contain too many sentences, the GPR scores could likewise be used to remove sentences.

The FRECCA algorithm also has features which are attractive for multi-document summarisation. Multi-document summarisation usually proceeds by pooling sentences from multiple documents into a single collection, and then summarising this collection. Unlike single document summarisation, where sentences with highly similar content are unlikely to be found, multiple documents that address similar or related topics are likely to contain similar or even identical sentences. This presents a difficulty for algorithms such as TextRank [Mihalcea and Tarau, 2004], since two sentences that are similar in content will have a similar PageRank score, and may therefore both be selected for the summary. One solution is to introduce a maximum threshold on the sentence similarity measure, and in the graph construction stage to only include links between sentences whose similarity does not exceed this threshold [Mihalcea and Tarau, 2005]. FRECCA avoids this problem because it clusters sentences prior to selecting sentences for the summary. If two or more sentences are similar, they are likely to appear in the same cluster; however, while their membership values to this cluster may be similar, they are unlikely to be identical, and the sentence with the highest value can simply be selected. Of course if we wish to add additional sentences (e.g., those with high GPR values), then we would still need to ensure that these were not too similar to the cluster centroids, and thresholding could be used to ensure this.

6.5 Discussion and Conclusions

The FRECCA algorithm was motivated by our interest in fuzzy clustering of sentence-level text, and the need for an algorithm which can accomplish this task based on relational input data. The empirical results show that the algorithm used in conjunction with the SSSE measure is able to achieve superior performance to both Spectral Clustering and ARCA algorithms when externally evaluated in hard clustering mode on challenging datasets of famous quotations, and applying the algorithm to a recent news article has demonstrated that the algorithm is capable of identifying overlapping clusters of semantically related sentences. Comparisons with the ARCA algorithm on each of these datasets suggest that FRECCA used in conjunction with the SSSE measure is capable of identifying softer clusters than ARCA, without sacrificing performance as evaluated by external measures.

FRECCA has a two attractive features. Firstly, based on empirical observations, it is not sensitive to the initialisation of cluster membership values, with repeated trials on all datasets converging to exactly the same values, irrespective of initialisation. This is in stark contrast to *k*-Means and Gaussian mixture approaches, which tend to be highly sensitive to initialisation. Secondly, the algorithm appears to be able to converge to an appropriate number of clusters, even if the number of initial clusters was set very high. For example, on the 50-Quotes dataset the final number of clusters was never greater than five (there were five actual classes in the dataset), and on the news article the algorithm converged to five clusters, which appears reasonable given the length, breadth, and general nature of the article.

The major disadvantage of the algorithm is its time complexity. As discussed in Section 6.1, PageRank must be applied to each cluster in each EM cycle, and this can lead to long convergence times if the problem involves a large number of objects and/or clusters. While the convergence time can be reduced significantly by initialising membership values with 0/1 values obtained from applying some inexpensive hard clustering algorithm, this still does not allow the algorithm to scale well to large datasets. The strength of FRECCA lies in its ability to identify fuzzy clusters, and if the objective is to perform only hard clustering, then a less costly algorithm such as Spectral Clustering should be preferred.

Although the algorithm has been applied to relational data, it can also be applied to attribute data. This might be done by first calculating pairwise distances between pairs of attribute vectors using some suitable distance measure (e.g., Euclidean, Mahalanobis, etc.), and then converting these distances to similarities by passing them through a suitable monotone decreasing function. In contrast with prototype-based algorithms such as Fuzzy *c*-Means, which can only discover compact (i.e., convex) clusters, FRECCA—because it is based on eigenvector centrality—is inherently capable of identifying non-compact clusters.

Further to the above, density estimation based on Gaussians is notoriously difficult in high-dimensional input spaces due to the curse of dimensionality [Bellman, 1961], and in particular, the tendency of data points to be increasingly concentrated towards the boundaries of the input space as the number of dimensions increases [Hastie *et al.*, 2001; Bishop, 1995]. This means that in high dimensional spaces, models based on Gaussian mixtures will almost certainly break down on account of the fact that these models are parameterised by the means of the mixture components, and these means will be located far from the majority of the probability mass. Because graph-based centrality measures use a richer input (i.e., the complete set of pairwise similarities), this, together with the recursive nature of eigenvector centrality, means that graph-based models should be better able to deal with the data sparsity inherent in high dimensional input spaces.

A second purpose of this chapter was to evaluate the SSSE similarity measure *in vivo* on sentence clustering tasks. The results on the two famous quotations datasets clearly show that the use of the SSSE measure consistently leads to better performance than that due to the basic Li measure, irrespective of which of the three clustering algorithms is used. This is almost certainly due to the fact that WSD used in conjunction with synonym expansion leads to a richer semantic context in which to compare sentences, ultimately resulting in improved clustering performance.

Chapter 7

Conclusion and Future Work

The aim of this research was to develop computational linguistic techniques for sentence-level text processing. This was motivated by the belief that successfully being able to capture the interrelationships between sentence-level text fragments would lead to an increase in the breadth and scope of problems to which clustering, classification, and other text mining activities can successfully be applied. Three specific questions were posed: *Can sentence similarity measurement be improved through incorporating WSD and context expansion*?, *Can WSD performance be improved by better utilising the context provided by the surrounding words?* and *Can a relational clustering algorithm be devised that is better able to capture the complex and subtle interrelationships between text objects at the sentence level?* These three research questions were gradually answered during the process of designing, developing and implementing the approaches described in Chapter 3 through to 6. While we believe that the research makes a significant contribution to the body of knowledge in these areas, the field continues to evolve rapidly, and new problems and challenges continue to emerge. The first part of this chapter summarises the key contributions of the thesis. We then identify some promising directions for future work.

7.1 Research Contributions

The thesis has made contributions in a number of areas, which we now summarise.

7.1.1 Sentential Word Importance

Chapter 3 introduced the notion of sentential word importance, and described how a measure of the relative importance of words in a given sentence can be determined using graph centrality measures. Although it was shown how the resulting word importance values could be incorporated within two well-known sentence similarity measures, the method is generic, and in principle can be incorporated into other sentence similarity measures as well. The results that were presented showed that while there appears to be some improvement resulting from the incorporation of word importance values, *in vitro* testing on standard (classification) datasets such as the MSRP and RTE datasets require a classification threshold to be set, and the classification performance can be sensitive to this value. This points to the importance of *in vivo* testing. The use of sentential word importance can also be used in other contexts; for example, in determining the order of disambiguation on WSD tasks.

7.1.2 Word Sense Disambiguation

Chapter 4 presented a new similarity-based WSD algorithm. Unlike conventional similarity-based methods, which are based on measuring pairwise similarity between *words*, and disambiguate words individually, usually without considering the senses assigned to surrounding words, the proposed approach is based on measuring semantic similarity between *sentences*, and progressively considering the senses assigned to surrounding disambiguated words. This enables the method to utilise a higher degree of semantic information contained in the surrounding context. The method has much lower complexity than graph-based methods, utilises the whole context of both surrounding words and the word-senses (glosses), yet performs comparably to, and in many cases exceeds the performance of graph-based approaches. It was also shown how performance can be further improved by incorporating a preliminary step in which the sentential word importance of words within the original text fragment is estimated, thereby providing an ordering that can be used to determine the sequence in which words should be disambiguated. Empirical results demonstrate that incorporating such an ordering clearly affects the disambiguation performance, leading to improved performance over a simple left-to-right ordering. Results have shown that the method performs favourably against state-of-the-art unsupervised WSD methods, as evaluated through both stand-alone (*in vitro*) and end-to-end (*in vivo*) evaluation models on a number of standard datasets.

7.1.3 Sentence Similarity using WSD and Synonym Expansion

Chapter 5 presented the third contribution of this thesis, which is a new sentence similarity measure that uses WSD and synonym expansion to provide an enriched semantic context, thus enabling a more accurate estimate of semantic similarity between two sentences. While the idea of incorporating WSD into sentence similarity measurement was recently explored in Ho *et al.* (2010), and the idea of context expansion was used in Feng *et al.* (2008) and Zhao *et al*. (2006), the method presented in this thesis is the first attempt to combine these ideas. What is important to stress is that WSD and synonym expansion are not independent: the method that has been presented operates by expanding semantic context in the direction indicated by the disambiguation; that is, the semantic context it is not expanded blindly, but is focused in the direction of the semantic context provided by the sense-assigned meanings of the original words. Results have shown that incorporating WSD and synonym expansion lead to better results than other methods recently reported in the literature, as evaluated *in vitro* using three benchmark datasets. Moreover, evaluating the method *in vivo* on a range of sentence clustering tasks, and using a range of clustering algorithms, demonstrated that the method yields superior clustering performance to that of a basic similarity measure. Importantly, this improvement is gained with very little increase in computational cost.

7.1.4 Fuzzy Relational Clustering

Chapter 6 introduced a novel fuzzy relational clustering algorithm. Inspired by the mixture model approach, the algorithm operates by modelling that data as combination of components. However, unlike conventional mixture models, which operate in a Euclidean space and use a likelihood function parameterised by the means and covariances of Gaussian components, the algorithm abandons use of any explicit density model (e.g., Gaussian) for representing clusters. Instead, a graph representation is used, in which nodes represent objects, and weighted edges represent the similarity between objects (i.e. sentences). By applying the PageRank algorithm to each cluster, and interpreting the PageRank score of an object within some cluster as a likelihood, the Expectation-Maximisation (EM) framework is then used to determine the model parameters (i.e., cluster membership values and mixing coefficients). The result is a fuzzy relational clustering algorithm which is generic in nature, and can be applied to any domain in which the relationship between objects is expressed in terms of pairwise similarities. Results of applying the algorithm to several sentence clustering tasks have shown in its performance to be superior to that of both Spectral Clustering and the ARCA algorithm. Applying the algorithm to a recent news article has demonstrated that the algorithm is capable of identifying overlapping clusters of semantically related sentences, and therefore of potential use in a variety of text processing tasks including document summarisation, and text mining of a more general nature.

7.2 Future Directions

The thesis has been concerned with developing computational linguistic techniques that can be used for text mining. One of these has been the development of the FRECCA algorithm for fuzzy relational clustering. While we have shown how the algorithm is of use in identifying fuzzy clusters of sentences, one obvious direction for future research would be to apply the algorithm to real text mining tasks. For example consider the task of query-directed text mining [Crabtree *et al.*, 2006], where the specific objective might be to discover some novel information from a set of documents initially retrieved in response to some query. Clustering the sentences of those documents is expected to provide the semantically meaningful nuggets of knowledge that are of interest, and intuitively we would expect at least one of the clusters to be closely related to the concepts described by the query terms; however, other clusters may contain information pertaining to the query in some way hitherto unknown to us, and in such a case we would have successfully mined new information. Another possible application is the rapidly growing area of opinion mining [Zhai *et al.*, 2011; Jia *et al.*, 2010].

While activities such as query-directed text mining are of interest in their own right, embedding clustering algorithms such as FRECCA within such tasks also has the dual purpose of evaluating the algorithm *in vivo*; i.e., on real tasks. In fact, *in vivo* testing and comparison of fuzzy clustering algorithms is particularly important in the case of fuzzy relational clustering due to the lack of unsupervised clustering criteria available. While a variety of unsupervised methods are available for evaluating centroid-based fuzzy clusterings, the only unsupervised measures currently available for relational methods are the Partition Coefficient (PC) [Bezdek, 1974] and closely related Partition Entropy Coefficient (PE) [Bezdek, 1975], and these are somewhat unsatisfying since they only provide a measure of the crispness of clustering.

A possible extension of the FRECCA algorithm is to extend it to perform hierarchical clustering. The concepts present in natural language documents usually display some type of hierarchical structure, and the ability to identify hierarchical relations of fuzzy clusters would be of use in activities such as automatically identifying concept hierarchies.

The Sentence Similarity using Synonym Expansion (SSSE) method that we have contributed is based on combining WSD and synonym expansion to provide a richer semantic context. Intuitively, the richer the semantic context, the better the performance we would expect of the measure. While in this thesis we have used *synonym* expansion, it may also be possible to consider other semantic relations such a hypernymy, meronymy, etc.

The effectiveness of the sentence similarity measure clearly also depends on the quality of the lexical resource. WordNet has been criticized as being limited in regards to its coverage [Achananuparp *et al.*, 2008], and exploring the feasibility of using larger knowledge bases such as Wikipedia as either a replacement for, or an adjunct to, WordNet may also lead to further improvement in disambiguation performance.

Sentence similarity measures depend on suitable word-to-word similarity measures. The sentence similarity measure we have proposed is orthogonal to the actual word-to-word similarity measure used, and new word-to-word similarity measures can easily be incorporated into the method, and may lead to further improvement.

Finally, while the idea of combining WSD and synonym expansion has been implemented within a reduced vector space model using the basic Li measure, the same idea can in principle also be implemented using other sentence representation schemes such as Latent Semantic Indexing (LSI). There is much current interest in the area of sentence similarity measurement, and we urge other researchers to explore incorporating these ideas into their own measures.

Appendix A

Benchmark Datasets for WSD and Sentence Similarity Experiments

This Appendix shows excerpts of the benchmark datasets used in our experiments in Chapters 3, 4, and 5.

SemCor for WSD Experiments

Table A.1: An excerpt of SemCor dataset representing the text fragment: “*Only a relative handful of such report was received, ‘the jury said’, considering the widespread interest in the election, the number of voters and the size of this city*”.

```
<contextfile concordance=brown>
<context filename=br-a01 paras=yes>

…

<p pnum=4>
<s snum=4>
<punc>``</punc>
<wf cmd=done pos=RB lemma=only wnsn=1 lexsn=4:02:02::>Only</wf>
<wf cmd=ignore pos=DT>a</wf>
<wf cmd=done pos=JJ lemma=relative wnsn=1 lexsn=3:00:00::>relative</wf>
<wf cmd=done pos=NN lemma=handful wnsn=1 lexsn=1:23:01::>handful</wf>
<wf cmd=ignore pos=IN>of</wf>
```

```
<wf cmd=done pos=JJ lemma=such wnsn=0 lexsn=5:00:01:specified:00>such</wf>
<wf cmd=done pos=NN lemma=report wnsn=3 lexsn=1:10:00::>reports</wf>
<wf cmd=done pos=VB ot=notag>was</wf>
<wf cmd=done pos=VB lemma=receive wnsn=2 lexsn=2:30:01::>received</wf>
<punc>"</punc>
<punc>,</punc>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=NN lemma=jury wnsn=1 lexsn=1:14:00::>jury</wf>
<wf cmd=done pos=VB lemma=say wnsn=1 lexsn=2:32:00::>said</wf>
<punc>,</punc>
<punc>``</punc>
<wf cmd=done pos=VB lemma=consider wnsn=4 lexsn=2:32:00::>considering</wf>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=JJ lemma=widespread wnsn=1
lexsn=5:00:00:general:00>widespread</wf>
<wf cmd=done pos=NN lemma=interest wnsn=1 lexsn=1:09:00::>interest</wf>
<wf cmd=ignore pos=IN>in</wf>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=NN lemma=election wnsn=1 lexsn=1:04:01::>election</wf>
<punc>,</punc>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=NN lemma=number wnsn=2 lexsn=1:23:00::>number</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=voter wnsn=1 lexsn=1:18:00::>voters</wf>
<wf cmd=ignore pos=CC>and</wf>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=NN lemma=size wnsn=1 lexsn=1:07:00::>size</wf>
<wf cmd=done pos=RB ot=notag>of_this</wf>
<wf cmd=done pos=NN lemma=city wnsn=1 lexsn=1:15:00::>city</wf>
<punc>"</punc>
<punc>.</punc>
</s>
</p>

…

</context>
</contextfile>
```

Senseval-2 and Senseval-3 for WSD Experiments

Table A.2: An excerpt of Senseval-3 English all-words labeled test dataset. The used Senseval corpora were converted into SemCor format and annotated with WordNet 3.0 senses.

```
<context filename=d000 source=senseval3>
…
<s snum=8>
<wf cmd=ignore pos=PRP>It</wf>
<wf cmd=ignore pos=VBD lemma=be>was</wf>
<wf cmd=done id=d000.s011.t002 pos=VBN lemma=blur wnsn=6
lexsn=2:30:00::>blurred</wf>
<punc>,</punc>
<wf cmd=ignore pos=IN>after</wf>
<wf cmd=ignore pos=CD>two</wf>
<wf cmd=done id=d000.s011.t007 pos=NNS lemma=hour wnsn=1
lexsn=1:28:00::>hours</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done id=d000.s011.t009 pos=JJ lemma=steady wnsn=1
lexsn=3:00:00::>steady</wf>
<wf cmd=done id=d000.s011.t010 pos=NN lemma=drinking wnsn=2
lexsn=1:04:01::>drinking</wf>
<punc>,</punc>
<wf cmd=ignore pos=CC>but</wf>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done id=d000.s011.t014 pos=NN lemma=occasion wnsn=1
lexsn=1:11:00::>occasion</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=ignore pos=PRP>it</wf>
<wf cmd=done id=d000.s011.t017 pos=VBD lemma=come_back wnsn=1
lexsn=2:30:00::>came_back</wf>
<wf cmd=ignore pos=TO>to</wf>
<wf cmd=ignore pos=PRP>him</wf>
<punc>.</punc>
</s>
…
</context>
```

MSRP for Sentence Similarity Experiments

Table A.3: An excerpt of MSRP test dataset. A binary values 1/0 (true/false) judgment as to whether one text-fragment was a valid paraphrase of the other.

Annotated Values	Pair of Text-fragments
…	
1	"**T$_1$**": The settling companies would also assign their possible claims against the underwriters to the investor plaintiffs, he added. "**T$_2$**": Under the agreement, the settling companies will also assign their potential claims against the underwriters to the investors, he added.
0	"**T$_1$**": Air Commodore Quaife said the Hornets remained on three-minute alert throughout the operation. "**T$_2$**": Air Commodore John Quaife said the security operation was unprecedented.
1	"**T$_1$**": A Washington County man may have the countys first human case of West Nile virus, the health department said Friday. "**T$_2$**": The countys first and only human case of West Nile this year was confirmed by health officials on Sept. 8.
1	"**T$_1$**": Moseley and a senior aide delivered their summary assessments to about 300 American and allied military officers on Thursday. "**T$_2$**": General Moseley and a senior aide presented their assessments at an internal briefing for American and allied military officers at Nellis Air Force Base in Nevada on Thursday.
0	"**T$_1$**": The broader Standard & Poor's 500 Index <.SPX> was 0.46 points lower, or 0.05 percent, at 997.02. "**T$_2$**": The technology-laced Nasdaq Composite Index .IXIC was up 7.42 points, or 0.45 percent, at 1,653.44.
…	

RTE for Sentence Similarity Experiments

Table A.4: An excerpt of RTE-3 test dataset. A tags <t>, <h> and entailment=”YES/No” representing the text, hypothesis and annotators answer of entailed or not respectively.

```
<?xml version="1.0" encoding="UTF-8" ?>
<entailment-corpus>

…

<pair id="94" entailment="YES" task="IE" length="short" >
<t>Techwood Homes was demolished in 1995 before the 1996 Summer Olympics. It and
neighboring Clark Howell Homes are now a mixed-use area called Centennial Place.</t>
<h>Techwood Homes was situated next to Clark Howell Homes.</h>
</pair>
<pair id="95" entailment="YES" task="IE" length="short" >
<t>After delivering her patients, the ship sailed 26 May 1946 for the atomic tests at Bikini
Atoll, and after providing medical services during the series of nuclear blasts during
Operation Crossroads, she returned to Seattle 15 August 1946.</t>
<h>The atomic tests at Bikini Atoll took place in 1946.</h>
</pair>
<pair id="96" entailment="YES" task="IE" length="short" >
<t>Lauren landed her first job performing in the national touring company of the musical
"Dreamgirls". Loraine would often step in for her sister in "Dreamgirls", in which they
played the role of "Mimi Marquez".</t>
<h>Loraine took part in the musical "Dreamgirls".</h>
</pair>
<pair id="97" entailment="YES" task="IE" length="short" >
<t>From the Illinois side of the bridge, signs marked "HISTORIC ROUTE 66 SPUR"
take travellers to the Illinois side of the bridge and a "HISTORIC ROUTE 66" sign marks
the Missouri side of the bridge.</t>
<h>Illinois borders on Missouri.</h>
</pair>
<pair id="98" entailment="NO" task="IE" length="short" >
<t>In the May 2005 general election Michael Howard failed to unseat the Labour
Government, although the Conservatives did gain 33 seats, playing the most significant
role in reducing Labour's majority from 167 to 66.</t>
<h>The Labour lost the majority in the May 2005 election.</h>
</pair>
<pair id="99" entailment="NO" task="IE" length="short" >
<t>In the May 2005 general election Michael Howard failed to unseat the Labour
```

Government, although the Conservatives did gain 33 seats, playing the most significant
role in reducing Labour's majority from 167 to 66.</t>
<h>Michael Howard was part of the Labour Government.</h>
</pair>

…

</entailment-corpus>

Appendix B

Famous Quotations Datasets

This Appendix shows the famous quotations datasets that used in our clustering experiments in Chapters 6.

50-Quotes Dataset

Table B.1: 50-Quotes dataset.

Knowledge

1. Our knowledge can only be finite, while our ignorance must necessarily be infinite.
2. Everybody gets so much common information all day long that they lose their commonsense.
3. Little minds are interested in the extraordinary; great minds in the commonplace.
4. Pocket all your knowledge with your watch and never pull it out in company unless desired.
5. Knowledge is of two kinds; we know a subject ourselves, or we know where we can find information upon it.
6. As we acquire more knowledge, things do not become more comprehensible, but more mysterious.
7. The learned is happy, nature to explore, the fool is happy, that he knows no more.
8. One of the greatest joys known to man is to take a flight into ignorance in search of knowledge.
9. Our knowledge is a receding mirage in an expanding desert of ignorance.
10. The specialist is a man who fears the other subjects.

Marriage

11. A husband is what is left of a lover, after the nerve has been extracted.
12. Marriage has many pains, but celibacy has no pleasures.
13. The woman cries before the wedding; the man afterward.
14. A rich widow weeps with one eye and signals with the other.
15. A wise woman will always let her husband have her way.
16. The calmest husbands make the stormiest wives.

17. Married couples who love each other tell each other a thousand things without talking.
18. Seldom, or perhaps never, does a marriage develop into an individual relationship smoothly and without crises; there is no coming to consciousness without pain.
19. A woman must be a genius to create a good husband.
20. A deaf husband and a blind wife are always a happy couple.

Nature

21. I have called this principle, by which each slight variation, if useful, is preserved, by the term natural selection.
22. Nature is reckless of the individual; when she has points to carry, she carries them.
23. I wanted to say something about the universe; there's God, angels, plants and horseshit.
24. The course of nature is the art of God.
25. From the intrinsic evidence of His creation, the Great Architect of the Universe now begins to appear as a pure mathematician.
26. Nature, with equal mind, sees all her sons at play, sees man control the wind, the wind sweep man away.
27. There is one glory of the sun and another glory of the moon and another glory of the stars for one star differeth from another star in glory.
28. The mastery of nature is vainly believed to be an adequate substitute for self mastery.
29. When I first open my eyes upon the morning meadows and look out upon the beautiful world, I thank god I am alive.
30. The beauty of the world and the orderly arrangement of everything celestial makes us confess that there is an excellent and eternal nature, which ought to be worshiped and admired by all mankind.

Peace

31. They sicken of the calm who know the storm.
32. When fire and water are at war it is the fire that loses.
33. Peace is a virtual, mute, sustained victory of potential powers against probable greeds.
34. We are each gifted in a unique and important way, it is our privilege and our adventure to discover our own special light.
35. I prefer the most unfair peace to the most righteous war.
36. War is an invention of the human mind, the human mind can invent peace.
37. The more we sweat in peace the less we bleed in war.
38. To be prepared for war is one of the most effectual means of preserving peace.
39. There is no such thing as inner peace, there is only nervousness and death.
40. Once you hear the details of victory, it is hard to distinguish it from a defeat.

Food

41. Food is an important part of a balanced diet.
42. To eat well in England you should have breakfast three times a day.
43. Dinner, a time when one should eat wisely but not too well, and talk well but not too wisely.
44. Hunger is not debatable.
45. To get the best results, you must talk to your vegetables.
46. A good meal ought to begin with hunger.
47. To a man with an empty stomach, food is god.
48. There is no such thing as a pretty good omelette.
49. Fish, to taste right, must swim three times in water, in butter and in wine.
50. At the end of every diet, the path curves back toward the trough.

211-Quotes Dataset

Table B.2: 211-Quotes dataset.

Politics

1. The fact that a reactionary can sometimes be right is a little less recognized that the fact that a liberal can be ...
2. Any woman who understands the problems of running a home will be nearer to understanding the problems of running a country.
3. If we cannot now end our differences, at least we can help make the world safe for diversity.
4. The great nations have always acted like gangsters, and the small nations like prostitutes.
5. The heaviest penalty for deciding to engage in politics is to be ruled by someone inferior to yourself.
6. The Labour Party is going about the country stirring up apathy.
7. Ultimately politics in a democracy reflects values much more than it shapes them.
8. A question which can be answered without prejudice to the government is not a fit question to ask.
9. The more you read about politics, the more you got to admit that each party is worse than the other.
10. Honest statesmanship is the wise employment of individual meannesses for the public good.
11. The Republicans have their splits right after election and Democrats have theirs just before an election.

Music

12. Let us not forget that the greatest composers were also the greatest thieves, they stole from everyone and everywhere.
13. A legend is an old man with a cane known for what he used to do.
14. A symphony is a stage play with the parts written for instruments instead of for actors.
15. Classic music is th' kind that we keep thinkin'll turn into a tune.
16. Discord occasions a momentary distress to the ear, which remains unsatisfied, and even uneasy, until it hears something better.
17. Canned music is like audible wallpaper.
18. I do not mind what language an opera is sung in so long as it is the language I don't understand.
19. It is the best of all trades to make songs, and the second best to sing them.
20. Jazz is about the only form of art existing today in which there is freedom of the individual without the loss of group contact.
21. The Sonata is an essentially dramatic art form, combining the emotional range in vivid presentation of a full-size stage drama with the terseness of a short story.
22. Twelve Highlanders and a bagpipe make a rebellion.
23. Sentimentally I am disposed to harmony; but organically I am incapable of a tune.
24. Conductors must give unmistakable and suggestive signals to the orchestra, not choreography to the audience.
25. How wonderful opera would be if there were no singers.
26. The function of pop music is to be consumed.
27. Harpists spend half their life tuning and the other half playing out of tune.
28. The musician who always plays on the same string, is laughed at.

Education

29. Learning makes a man fit company for himself.
30. I forget what I was taught. I only remember what I have learnt.
31. If you educate a man you educate a person, but if you educate a woman, you educate a family.
32. The true teacher defends his pupils against his own personal influence.
33. The university is the last remaining platform for national dissent.
34. The investigation of the meaning of words is the beginning of education.
35. Schoolmasters and parents exist to be grown out of.
36. Creative minds have always been known to survive any kind of bad training.
37. Fathers send their sons to college either because they went to college, or because they didn't.
38. Whatever is good to know is difficult to learn.
39. The difference between genius and stupidity is that genius has its limits.
40. A kindergarten teacher is a woman who knows how to make little things count.
41. Schoolhouses are the republican line of fortifications.
42. What greater or better gift can we offer the republic than to teach and instruct our youth.

Success

43. The world belongs to the enthusiast who keeps cool.
44. Men at some time are masters of their fates.
45. The secret of success is constancy to purpose.
46. Survival is triumph enough.
47. The secret of all victory lies in the organization of the non obvious.
48. The conditions of conquest are always easy. We have but to toil awhile, endure awhile, believe always, and never turn back.
49. The very first step towards success in any occupation is to become interested in it.
50. Four steps to achievement: plan purposefully, prepare prayerfully, proceed positively, pursue persistently.
51. Always aim for achievement, and forget about success.
52. The way to rise is to obey and please.
53. We would accomplish many more things if we did not think of them as impossible.
54. Faith that the thing can be done is essential to any great achievement.
55. The will to conquer is the first condition of victory.
56. Man never rises to great truths without enthusiasm.
57. The method of the enterprising is to plan with audacity and execute with vigor.
58. Great minds have purposes, others have wishes.

Work

59. Give the labourer his wage before his perspiration be dry.
60. Everything considered, work is less boring than amusing oneself.
61. A good horse should be seldom spurred.
62. It is impossible to enjoy idling thoroughly unless one has plenty of work to do.
63. The test of a vocation is the love of the drudgery it involves.
64. Beware all enterprises that require new clothes.
65. Employment is nature physician, and is essential to human happiness.
66. Love of bustle is not industry.
67. A task becomes a duty from the moment you suspect it to be an essential part of that integrity which alone entitles a man to assume responsibility.
68. The worst crime against working people is a company which fails to operate at a profit.
69. The effectiveness of work increases according to geometric progression if there are no interruptions.
70. Do your duty until it becomes your joy.
71. In all human affairs there are efforts, there are results, and the strength of the effort is the

measure of the result.
72. Any man who has had the job I've had and didn't have a sense of humor wouldn't still be here.
73. Handle your tools without mittens.

Forgiveness
74. Reconciliation is more beautiful than victory.
75. Forgetting is the cost of living cheerfully.
76. Forgiveness is the key to action and freedom.
77. Life appears to me too short to be spent in nursing animosity or registering wrong.
78. Anger repressed can poison a relationship as surely as the crudest words.
79. I know now that patriotism is not enough; I must have no hatred and bitterness toward anyone.
80. Stretch out your hand Let no human soul wait for a benediction.
81. Courage and clemency are equal virtues.
82. Who understands much, forgives much.
83. The heart has always the pardoning power.
84. The whole human race loses by every act of personal vengeance.
85. Forgiveness is the sweetest revenge.
86. Blessed are those who can give without remembering and take without forgetting.
87. It is strange what a contempt men have for the joys that are offered them freely.
88. The heart is great which shows moderation in the midst of prosperity.

Experience
89. Let weakness learn meekness.
90. Experience has two things to teach, the first is that we must correct a great deal; the second that we must not correct too much.
91. Everything happens to everybody sooner or later if there is time enough.
92. Life is like playing a violin solo in public, and learning the instrument as one goes on.
93. Experience enables you to recognize a mistake when you make it again.
94. From error to error one discovers the entire truth.
95. Experience is the extract of suffering.
96. All that I know I learned after I was thirty.
97. Many of the insights of the saint stem from his experience as a sinner.
98. A proverb is no proverb to you till life has illustrated it.

Health
99. God heals and the doctor takes the fee.
100. Diets are for people who are thick and tired of it.
101. Whenever I feel like exercise, I lie down until the feeling passes.
102. The art of medicine consists of amusing the patient while nature cures the disease.
103. Your medical tests are in. You're short, fat, and bald.
104. The trouble with jogging is that, by the time you realize you're not in shape for it, it's too far to walk back.
105. There must be something to acupuncture - after all, you never see any sick porcupines.
106. Health is not a condition of matter, but of Mind.
107. Health lies in labor, and there is no royal road to it but through toil.
108. Refuse to be ill, never tell people you are ill; never own it to yourself.
109. Illess is one of those things which a man should resist on principle at the onset.

Law
110. Law and order is one of the steps taken to maintain injustice.
111. Lawyers and painters can soon change white to black.
112. A successful lawsuit is the one worn by a policeman.

113. To some lawyers, all facts are created equal.
114. Lawyers are the only persons in whom ignorance of the law is not punished.
115. When you have no basis for an argument, abuse the plaintiff.
116. Litigant a person about to give up his skin for the hope of retaining his bone.
117. A judge is a law student who marks his own examination papers.
118. I know of no method to secure the repeal of bad or obnoxious laws so effective as their stringent execution.
119. Divorce is a game played by lawyers.
120. In England, justice is open to all, like the Ritz Hotel.
121. You are remembered for the rules you break.
122. Any fool can make a rule, and every fool will mind it.
123. When you have no basis for an argument, abuse the plaintiff.
124. A jury consists of twelve persons chosen to decide who has the better lawyer.
125. Litigant, a person about to give up his skin for the hope of retaining his bone.

Spirituality

126. The spiritual life does not remove us from the world but leads us deeper into it.
127. Faith is the evidence of things not seen.
128. Live your beliefs and you can turn the world around.
129. The strength of a man consists in finding out the way in which God is going, and going in that way too.
130. Millions of angels are at God 's command.
131. Time spent on the knees in prayer will do more to remedy heart strain and nerve worry than anything else.
132. Soul appears when we make room for it.
133. I have lived to thank God that all my prayers have not been answered.
134. There are no atheists on turbulent airplanes.
135. Straight praying is never born of crooked conduct.
136. Religion without humanity is a poor human stuff.
137. We cannot ask in behalf of Christ what Christ would not ask Himself if He were praying.
138. It is impossible to lose your footing while on your knees.
139. The Vatican is against surrogate mothers; good thing they didn't have that rule when Jesus was born.
140. Fo those leaning on the sustaining infinite, today is big with blessings.
141. Jesus makes the bitterest mouthful taste sweet.
142. The person who has a firm trust in the Supreme Being is powerful in his power, wise by his wisdom, happy by his happiness.

Marriage

143. A husband is what is left of a lover, after the nerve has been extracted.
144. A rich widow weeps with one eye and signals with the other.
145. A wise woman will always let her husband have her way.
146. Marriage has many pains, but celibacy has no pleasures.
147. The calmest husbands make the stormiest wives.
148. Married couples who love each other tell each other a thousand things without talking.
149. Seldom, or perhaps never, does a marriage develop into an individual relationship smoothly and without crises; there is no coming to consciousness without pain.
150. A woman must be a genius to create a good husband.
151. A deaf husband and a blind wife are always a happy couple.
152. The woman cries before the wedding; the man afterward.
153. Bigamy is having one husband too many, Monogamy is the same.
154. Whoso findeth a wife findeth a good thing.

155. The young man who wants to marry happily should pick out a good mother and marry one of her daughters - any one will do.
156. Polygamy, an endeavour to get more out of life than there is in it.
157. Human love is often but the encounter of two weaknesses.
158. Romance without finance is no good.
159. The toughest thing about being a housewife is you have no place to stay home from.

Food

160. Food is an important part of a balanced diet.
161. At the end of every diet, the path curves back toward the trough.
162. Hunger is not debatable.
163. To eat well in England you should have breakfast three times a day.
164. To get the best results, you must talk to your vegetables.
165. Dinner, a time when . . . one should eat wisely but not too well, and talk well but not too wisely.
166. To a man with an empty stomach, food is god.
167. A good meal ought to begin with hunger.
168. There is no such thing as a pretty good omelette.
169. Fish, to taste right, must swim three times - in water, in butter and in wine.
170. There is no such thing as a little garlic.
171. A hungry man is not a free man.
172. All happiness depends on a leisurely breakfast.
173. A gourmet is just a glutton with brains.
174. A good meal makes a man feel more charitable toward the whole world than any sermon.
175. Sacred cows make the best hamburger.
176. Strength is the ability to break a chocolate bar into four pieces with your bare hands - and then eat just one of those pieces.

Intelligence

177. Action is the real measure of intelligence.
178. Always be smarter than the people who hire you.
179. Being an intellectual creates a lot of questions and no answers.
180. Failure is simply the opportunity to begin again, this time more intelligently.
181. Genius develops in quiet places, character out in the full current of human life.
182. The public is wonderfully tolerant. It forgives everything except genius.
183. Intelligence without ambition is a bird without wings.
184. Small minds are concerned with the extraordinary, great minds with the ordinary.
185. There are no great limits to growth because there are no limits of human intelligence, imagination, and wonder.

Peace

186. There is no such thing as inner peace, there is only nervousness and death.
187. When fire and water are at war it is the fire that loses.
188. Peace is a virtual, mute, sustained victory of potential powers against probable greeds.
189. Once you hear the details of victory, it is hard to distinguish it from a defeat.
190. They sicken of the calm who know the storm.
191. We are each gifted in a unique and important way, it is our privilege and our adventure to discover our own special light.
192. I prefer the most unfair peace to the most righteous war.
193. To be prepared for war is one of the most effectual means of preserving peace.
194. War is an invention of the human mind, the human mind can invent peace.
195. The more we sweat in peace the less we bleed in war.

196. Once you hear the details of victory, it is hard to distinguish it from a defeat.
197. The Dove, on silver pinions, winged her peaceful way.

Money
198. A bank is a place that will lend you money if you can prove that you don't need it.
199. A billion here, a billion there, and pretty soon you're talking about real money.
200. A business that makes nothing but money is a poor business.
201. Wealth flows from energy and ideas.
202. The petty economies of the rich are just as amazing as the silly extravagances of the poor.
203. The darkest hour of any man 's life is when he sits down to plan how to get money without earning it.
204. Bankruptcy is a legal proceeding in which you put your money in your pants pocket and give your coat to your creditors.
205. The rich aren't like us, they pay less taxes.
206. Definition of the upper crust, A bunch of crumbs held together by dough.
207. Take care of the pence, and the pounds will take care of themselves.
208. Ah, take the Cash, and let the Credit go, Nor heed the rumble of a distant Drum.
209. Not greedy of filthy lucre.
210. All lasting business is built on friendship.
211. If you can count your money you do not have a billion dollars.

Bibliography

Abdalgader, K. and Skabar, A. 2010. Short-text similarity measurement using word sense disambiguation and synonym expansion. In *Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence*. (AI2010, Adelaide). Advances in Artificial Intelligence. 6464, 435-444.

Abdalgader, K. and Skabar, A. 2011. Unsupervised similarity-based word sense disambiguation using context vectors and sentential word importance. *ACM Transactions on Speech and Language Processing*. (**Accepted subject to satisfactory revision. Revised manuscript submitted on 5th of October 2011**).

Achananuparp, P., Hu, X., and Yang, C. 2009. Addressing the variability of natural language expression in sentence similarity with semantic structure of the sentences. In *Proceedings of PAKDD 2009*. Bangkok. 548-555.

Agirre, E. and Rigau, G. 1996. Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistic* (COLING΄96, Copenhagen, Denmark). 16-22.

Agirre, E. and Stevenson, M. 2006. Knowledge sources for WSD. In *Word Sense Disambiguation: Algorithms and Applications*, E. Agirre and P. Edmonds, Eds. Springer, New York, NY. 217-251.

Agirre, E. and Soroa, A. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of* *the Association for Computational Linguistics*, (EACL'09, Athens, Greece). 33–41.

Aliguyev, R.M. 2009. A New sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*. 36, 7764-7772.

Allen, J. 1995. *Natural language understanding*. The Benjamin / Cummings Publ., Amsterdam, Bonn, Sidney, Singapore, Tokyo, Madrid, 1995.

Atkinson-Abutridy, J., Mellish, C. and Aitken, S. 2004. Combining information extraction with genetic algorithms for text mining. *IEEE Intelligent Systems*. 19, 3, 22-30.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. ACM Press, New York.

Baeza-Yates, R. 2004. Challenges in the interaction of information retrieval and natural language processing. In *Proceedings of 5th International Conference on Computational Linguistics and Intelligent Text Processing* (CICLing 2004, Seoul, Corea). Lecture Notes in Computer Science, Springer. 2945, 445-456.

Ball, G. and Hall, D. 1967. A clustering technique for summarizing multivariate data. *Behavioural Science*. 12, 153-155.

Banerjee, S. and Pedersen, T. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. (IJCAI'03). 805-810.

Barzilay, R. and Elhadad, M. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*. 10-17.

Bates, M. J. 1986. Subject access in online catalogue: A design model. *Journal of the American Society for Information Science*. 37, 6, 357-376.

Bellman, R.E. 1961. *Adaptive control processes*. Princeton University Press.

Berger, A. and Lafferty, J. 1999. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval* (SIGIR'99). 222-229.

Bezdek, J.C. 1974. Cluster validity with fuzzy sets. *Journal of Cybernetics*. 3, 3, 58-72.

Bezdek, J.C. 1975. Mathematical models for systematics and taxonomy. In *Proceedings of the 8th International Conference in Numerical Taxonomy*. 143-166.

Bezdek, J.C. 1981. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York.

Bilotti, M.W., Ogilvie, P., Callan, J. and Nyberg, E. 2007. Structured retrieval for question answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '07). ACM, New York. 351-358.

Bishop, C. 1995. *Neural networks for pattern recognition*. Oxford University Press, Oxford.

Bolshakov, I. and Gelbukh, A. 2004. *Computational linguistics: Models, Resources, Applications*. IPN-UNAM-FCE, ISBN 970-36-0147-2.

Bonacich, P. 1972. Factoring and weighting approaches to clique identification. *Journal of Mathematical Sociology*. 2, 113-120.

Bonacich, P. 2007. Some unique properties of eigenvector centrality. *Social Networks*. 29, 555-564.

Brandes, U. and Erlebach, T. 2005. *Network analysis: methodological foundations*. Springer.

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*. 30, 107-117.

Brown, R. and Frederking, R. 1995. Applying statistical English language modeling to symbolic machine translation. In *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation* (TMI'95). 221-239.

Budanitsky, A. and Hirst, G. 2006. Evaluating Wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*. 32, 1, 13-47.

Burgess, C. 1998. From simple associations to the building blocks of language: modeling meaning in memory with the HAL model. *Behaviour Research Methods, Instruments & Computers*. 30, 188-198.

Carpuat, M. and Wu, D. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (EMNLP-CoNLL, Prague, Czech Republic). 61–72.

Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., de Marneffe, M., Ramage, D., Yeh, E. and Manning, C. 2007. *Learning alignments and leveraging natural logic*. In *Proceedings of ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. 165-170.

Chan, Y. S., Ng, H. T., and Chiang, D. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (Prague, Czech Republic). 33–40.

Charniak, E. 1984. *Introduction to artificial intelligence*. 2, Addison-Wesley.

Charniak, E. 1995. Natural language learning. *ACM Computing Surveys*. 27, 3, 17-319.

Charniak, E., Blaheta, D., Ge, N., Hall, K., Hale, J. and Johnson, M. 2000. Bllip 1987-89 WSJ corpus release 1. Tech. rep. LDC2000T43. *Linguistic Data Consortium* (Philadelphia, PA).

Chen, F., Han, K. and Chen, G. 2008. An approach to sentence selection based text summarization. In *Proceedings of IEEE TENCON02*. 489-493.

Coelho, T., Calado, P., Souza, L., Ribeiro-Neto, B. and Muntz, R. 2004. Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. 16, 4, 408-417.

Corsini, P., Lazzerini, F. and Marcelloni, F. 2005. A new fuzzy relational clustering algorithm based on the fuzzy C-Means algorithm. *Soft Computing*. 9, 439-447.

Cowie, J., Guthrie, J., and Guthrie, L. 1992. Lexical disambiguation using simulated annealing. In *Proceedings of the 14th International Conference on Computational Linguistic* (COLING΄92, Nantes, France). 359-365.

Crabtree, D., Andreae, P., and Gao, X. 2006. Query directed web page clustering. In *Proceedings of the International Conference on Web Intelligence WI'06*. 202-210.

Dagan, I. and Glickman, O. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *Learning Methods for Text Understanding and Mining Workshop*.

Dagan, I, Glickman, O. and Magnini, B. 2005. The PASCAL recognizing textual entailment challenge. In *Quinonero-Candela et al.*, editor, MLCW 2005. Springer-Verlag, LNAI, 3944, 177–190.

Dagan, I., Dolan, B., Giampiccolo, D. and Magnini, B. 2007. The Third PASCAL Recognizing Textual Entailment Challenge. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. 1-9.

Dempster, A.P., Laird, N.M. and Rubin, D.B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*. Series B (Methodological). 39, 1, 1-38.

Dolan, W, Chris Quirk, C, and Brockett, CV. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*. 350-356.

Duda, R.O., Hart, P. E. and Stork, D.G. 2001. *Pattern Classification*. 2nd edition. John Wiley & Sons.

Dunn, J.C. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*. 3, 3, 32-57.

Erkan, G. and Radev, D. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*. 22, 457-479.

Feldman, S. 1999. *NLP meets the jabberwocky*. Online, 23, 62-72.

Fellbaum, C. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge.

Feng, J., Zhou, Y.-M. and Martin, T. 2008. Sentence similarity based on relevance. In *Proceedings of the IPMU'08*. 832-839.

Fox, C. 1990. *Lexical analysis and stop list*. Prentice-Hall, Upper Saddle River, NJ.

Francis, W. and Kucera, H. 1964. Brown corpus manual: manual of information to accompany a standard corpus of present-day edited American English, for use with digital computers. *Brown University*, Providence, Rhode Island.

Francis, W. and Kucera, H. 1982. *Frequency analysis of English usage*. Houghton Mifflin, New York, NY.

Freeman, L. C. 1979. Centrality in social networks: conceptual clarification I. *Social Networks*. 1,3, 215–239.

Frey, B.J. and Dueck, D. 2007. Clustering by passing messages between data points. *Science*. 315, 972-976.

Frobenius, G. 1912. Ueber matrizen aus nicht negativen elementen. *Sitzungsberichte der Preussischen Akademie der Wissenschaften zu Berlin*. 456-477.

Gale, W. A., Church, K. and Yarowsky, D. 1992. A method for disambiguating word senses in a corpus. *Comput. Human*. 26, 415-439.

Gee, K. R. and Cook, D. J. 2005. Text classification using graph-encoded linguistic elements. In *Proceedings of FLAIRS Conference*. 487-492.

Geweniger, T., Zühlke, D., Hammer, B. and Villmann, T. 2009. Fuzzy variant of affinity propagation in comparison to median fuzzy c-Means. In *Proceedings of the 7th international Workshop on Advances in Self-Organizing Maps*. Springer-Verlag, Berlin, Heidelberg. 72-79.

Geweniger, T., Zühlke, D., Hammer, B. and Villmann, T. 2010. Median fuzzy C-Means for clustering dissimilarity data. *Neurocomputing*. 73, 7-9, 1109-1116.

Giampiccolo, D., Magnini, B., Dagan, I. and Dolan, B. 2007. The third pascal recognizing textual entailment challenge.

Goker, A. and Davies, J. (Eds.). 2009. Information Retrieval: Searching in the 21st Century. *London: Wiley*. 295 (220). ISBN: 978-0-470-02762-2.

Halliday, M. and Hasan, R. 1976. *Cohesion in English*. London: Longman.

Harabagiu, S. and Moldovan, D., 1998. Knowledge processing on an extended WordNet. In: *Fellbaum, C. (Ed.), WordNet – An Electronic Lexical Database*. MIT Press, Cambridge, MA. 379-406.

Harris, Z. 1954. Distributional structure. In J. J. Katz, editor*, The Philosophy of Linguistics*. Oxford University Press, New York. 26-47.

Hastie, T., Tibshirani, R. and Friedman, J. 2001. *The elements of statistical learning: data mining*. Inference and Prediction. Springer, New York.

Hathaway, R. J., Devenport, J. W. and Bezdek, J. C. 1989. Relational dual of the C-Means clustering algorithms. *Pattern Recognition*. 22, 2, 205-212.

Hathaway, R. J. and Bezdek, J. C. 1994. NERF C-Means: Non-Euclidean relational fuzzy clustering. Pattern Recognition. 27, 429-437.

Hatzivassiloglou, V., Klavans, J. L., Holcombe, M. L., Barzilay, R., Kan, M.-Y. and McKeown, K. R. 2001. SIMFINDER: A flexible clustering tool for summarization. In *NAACL Workshop on Automatic Summarization*. Association for Computational Linguistics. 41-49.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING 92*. 539-545.

Hickl, A. and Bensley, J. 2007. A discourse commitment-based framework for recognizing textual entailment. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. 171-176.

Hirst, G. and St-Onge, D. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In *WordNet: An Electronic Lexical Database*. C. Fellbaum, Ed. MIT Press, Cambridge, MA. 305–332.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*. Pittsburg, Pennsylvania. 268-275.

Ho, C., Murad, M., Abdul Kadir, R. and Doraisamy, S.C. 2010. Word sense disambiguation-based sentence similarity. In *Proceedings of the 23rd International Conference on Computational Linguistic, COLING 2010*. Beijing. 418-426.

Hotho, A., Nürnberger, A. and Paaß, G. 2005. A brief survey of text mining. *GLDV-Journal for Computational Linguistics and Language Technology*. 20, 19-62.

Ide, N. and Veronis, J. 1998. Word sense disambiguation: The state of the art. *Computational Linguistic*. 24, 1, 1-40.

Islam, A. and Inkpen, D. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transaction on Knowledge Discovery from Data (TKDD)*. 2, 2, 1-25.

Jelinek, F. 1999. *Statistical methods for speech recognition*. Cambridge, MA: MIT Press.

Jeon, M., Park, H. and Rosen, J.B. 2001. Dimensional reduction based on centroids and least squares for efficient processing of text data. In *Proceedings for the 1st SIAM International Workshop on text mining*. Chicago,LL.

Jia, W., Zhang, S., Xia, Y., Zhang, J. and Yu, H. 2010. A novel product features categorize method based on twice-clustering. In *Proceedings of the 2010 International Conference on Web Information Systems and Mining, WISM'10*. 1, 281-284.

Jiang, J.J., Conrath, D.W. 1997. *Semantic similarity based on corpus statistics and lexical taxonomy*. In *Proceedings of the 10th International Conference on Research in Computational Linguistics*. 19-33.

Jurafsky, D. and Martin, J. H. 2009. *Speech and language processing: An introduction to natural language processing, speech recognition, and computational linguistics*. 2nd edition. Prentice-Hall.

Kanerva, P., Kristofersson, J. and Holst, A. 2000. Random indexing of text samples for Latent Semantic Analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*. 1036, 2, 16429-16429.

Kaufman, L. and Rousseeuw, P. J. 1987. Clustering by means of Medoids. *Statistical Analysis based on the L_1 Norm, Y. Godge, eds*., North Holland/Elsevier, Amsterdam. 405-416.

Kaufman, L. and Rousseeuw, P. J. 1990. *Finding groups in data*. Wiley.

Kazakov, D., Manandhar, S., and Erjavec, T. 1999. Learning word segmentation rules for tag prediction. In S. Dzeroski & P. Flach (Eds.). Inductive Logic Programming. In *Proceedings of 9th International Workshop*, ILP-99.152-161. Berlin: Springer-Verlag.

Kilgarriff, A. and Rosenzweig, J. 2000. English SENSEVAL: Report and Results. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*. (LREC, Athens, Greece). 1239-1244.

Kilgarriff, A. and Palmer, M. 2000. Introduction to the special issue on Senseval. *Comput. Human*. 34, 1-2, 1–13.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*. 46, 5, 604-632.

Kosala, R. and Blockeel, H. 2000. Web mining research: A survey. *ACM SIGKDD Explorations Newsletter*. 2, 1, 1-15.

Krishnapuram, R., Joshi, A. and Liyu, Y. 1999. A fuzzy relative of the k-Medoids algorithm with application to web document and snippet clustering. In *Proceedings of the IEEE Fuzzy Systems Conference*. 1281-1286.

Krovetz, R. 1993. Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '93)*. Pittsburgh, Pennsylvania. 191-202.

Kyoomarsi, F., Khosravi, H., Eslami, E., Dehkordy, P.K and Tajoddin, A. 2008. Optimizing text summarization based on fuzzy logic. *Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE Computer Society*. 347-352.

Landauer, T. K., Foltz, P. W. and Laham, D. 1998. An introduction to latent semantic analysis. *Discourse Processes*. 25, 259-284.

Leacock, C. and Chodorow, M. 1998. Combining local context and WordNet similarity for word sense identification. In *C. Fellbaum (Ed.), Cambridge, Mass*. MIT Press, Chp. 11, 265-283.

Lee, D. and Seung, H. 2001. Algorithms for Non-Negative matrix factorization. *Advances in Neural Information Processing Systems*. 13, 556-562.

Lemaire, B. and Denhière, G. 2004. Incremental construction of an associative network from a corpus. In *Proceedings of the 26th Annual Meeting of the Cognitive Science*. Society (CogSci'2004). 825-830

Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on systems documentation*. (SIGDOC, Toronto, Canada). 24-26.

Li, Y., McLean, D., Bandar, Z., O'Shea, F. and Crockett, K. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. 18, 8, 1138-1150.

Li, Y., McLean, D., Bandar, Z., O'Shea, F. and Crockett, K. 2009. Pilot short text semantic similarity benchmark data set: Full Listing and Description. (http://www.mendeley.com).

Liddy, E. 1998. Enhanced text retrieval using natural language processing. *Bulletin of the American Society for Znformation Science*. 24, 4, 1P16.

Liddy, E. D. 2010. Natural language processing for information retrieval. *Encyclopedia of Library and Information Sciences*. 3rd Edition 3864-3873.

Lin, D. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*. Madison, Wisc. 296-304.

Litkowski, K. C. 2005. Computational lexicons and dictionaries. In *Encyclopedia of Language and Linguistics* (2nd ed.), K. R. Brown, Ed. Elsevier Publishers, Oxford, U.K. 753-761.

Luxburg, U.V. 2007. A tutorial on spectral clustering. *Statistics and Computing*. 17, 4, 395-416.

MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. 281-297.

Manning, C. D., Raghavan, P., Schütze, H. 2008. *Introduction to information retrieval*. Cambridge University Press, Cambridge.

Martinez, D. 2004. Supervised word sense disambiguation: Facing current challenges. Ph.D. dissertation. University of the Basque Country, Spain.

Metzler, D., Dumais, S. and Meek, C. 2007. Similarity measures for short segments of text. In *Proceedings of the 29th European Conference on Information Retrieval*. 4425, Springer, Heidelberg. 16-27.

Meila, M. and Shi, J. 2001. Learning segmentation by random walks. *Advances in Neural Information Processing Systems*. 14.

Mihalcea, R. and Moldovan, D. I. 1999. Automatic acquisition of sense tagged corpora. In A. N. Kumar & I. Russell (Eds.), In *Proceedings of the !helfth Znternational Florida AZ Research Society Conference*. 293-297. Menlo Park, CA: AAAI Press.

Mihalcea, R. and Moldovan, D. I. 2001. eXtended Word-Net: Progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*. (Pittsburgh, PA). 95-100.

Mihalcea, R. and Tarau, P. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP*. 404-411.

Mihalcea, R., Tarau, P. and Figa, E. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics* (COLING 2004), Geneva, Switzerland.

Mihalcea, R. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. (HLT/EMNLP, Vancouver, Canada). 411-418.

Mihalcea, R. and Tarau, P. 2005. An algorithm for language independent single and multiple document summarization. In *Proceedings of the International Joint Conference on Natural Language Processing* (IJCNLP). Korea.

Mihalcea, R., Corley, C., Strapparava, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of 21st National Conference on Artificial Intelligence*. 1, 775-780.

Miller, G., Leacock, C., Randee, T., and Bunker, R. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA workshop on Human Language Technology*. 303-308.

Mohler, M. and Mihalcea, R. 2009. Text-to-Text semantic similarity for automatic short answer grading. In *Proceedings of EC-ACL 2009*. Athens, Greece. 567-575.

Montoyo, A., Suarez, A., Rigau, G. and Palomar, M. 2005. Combining knowledge- and corpus-based word sense disambiguation methods. *Journal of Artificial Intelligence Research*. 23, 299-330.

Namburu, S.M., Tu, H., Luo, J., and Pattipati, K.R. 2005. Experiments on supervised learning algorithms for text categorization. *IEEE Aerospace Conference*. Big Sky, MT.

Naughton, M., Kushmerick, N. and Carthy, J. 2006. Clustering sentences for discovering events in news articles. In *Proceedings of ECIR*. 535-538.

Navigli, R. and Velardi, P. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI). 27, 7, 1075-88.

Navigli, R. 2008. A structural approach to the automatic adjudication of word sense disagreements. *Natural Language Engineering*. 14, 4, 547-573.

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*. (CSUR). 41, 2, 1-69.

Navigli, R. and Lapata, M. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI). 32, 4, 678-692.

Ng, A.Y., Jordan, M.I. and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*. 849-856.

Paice, C. D. 1996. Method for evaluation of stemming algorithms based on error counting. *Journal of the American Society for Information Science*. 47, 8, 632-649.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., and Dang, H. 2001. English tasks: all-words and verb lexical sample. In *Proceedings of ACL/SIGLEX* (Senseval-2, Toulouse, France). 21–24.

Palmer, M., Ng, H. T. and Dang, H. T. 2006. Evaluation of WSD systems. In *Word Sense Disambiguation: Algorithms and Applications*. E. Agirre and P. Edmonds, Eds. Springer, New York, NY. 75-106.

Patwardhan, S., Banerjee, S., and Pedersen, T. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics*. (CICLing'03, Mexico City). 241-257.

Pazienza, M. T., Pennacchiotti, M., and Zanzotto, F. M. 2005. Terminology extraction: an analysis of linguistic and statistical approaches. In *Knowledge Mining*. 185, 255-279.

Pedersen, T. 2008. Computational approaches to measuring the similarity of short contexts: a review of applications and methods. CoRR. Abs/0806.3787.

Perron, O. 1907. Zur theorie der matrices. *Mathematische Annalen*. 64, 2, 248-263.

Peter, F., Brown, J. C., Stephen, A. P., Vincent, J. P., Frederick, J., John, D. L., Robert, L. M., and Paul, S. R. 1990. A statistical approach to machine translation. *Computational Linguistics*. 16, 2, 79–85.

Pradhan, S., Loper, E., Dligach, D., and Palmer, M. 2007. Semeval-2007 Task-17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, (SemEval'07). 87-92.

Ponte, J. M. and Croft, W. B. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st international conference on research and development in information retrieval* (SIGIR'98). 275-281.

Porter, M. 1980. *An Algorithm for suffix stripping*. Program. 14, 3.

Rabiner, L. R. and Juang, B. H. 1986. An introduction to hidden markov models. *IEEE Magazine on Accoustics, Speech and Signal Processing*. 3, 1, 4-16.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. 1989. Development and application of a metric to semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*. 19, 1, 17-30.

Radev, D.R., Jing, H., Stys, M., and Tam, D. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management: AN International Journal*. 40, 919-938.

Ramage, D., Rafferty, A., Manning, C. 2006. Random walks for text semantic similarity. In *Proceedings of ACL-IJCNLP 2009*. 23-31.

Ramakrishnan, G., Jadhav, A., Joshi, A., Chakrabarti, S., and Bhattacharyya, P. 2003. Question answering via bayesian inference on lexical relations. In *Proceedings of the ACL workshop on Multilingual Summarization and Question Answering*. (MultiSumQA). 1–10.

Rand, W.M. 1971. Objective criteria for the evaluation of clustering methods. *American Statistical Association Journal*. 66, 338, 846-850.

Resnik, P. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. (IJCAI'95, Montreal, Canada) 1, 448-453.

Resnik, P. and Yarowsky, D. 1997. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?* (Washington, D.C.). 79–86.

Riabinin, Y. 2008. Recognizing textual entailment using logical inference: A survey of the pascal rte challenge. Online.

Rosenberg, A. and Hirschberg, J. 2007. V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the EMNLP 2007*. 410-20.

Rosenfield, R. 2000. Two decades of statistical language modeling: Where do we go from here?. In *Proceedings of the IEEE*. 88, 8, 1270-1278.

Roubens, M. 1978. Pattern classification problems and fuzzy sets. *Fuzzy Sets and Systems*. 1, 239-253.

Rubenstein, H. and Goodenough, J. B. 1965. Contextual correlates of synonymy. *Comm. ACM.* 8, 10, 627-633.

Ruspini, E. H. 1969. A new approach to clustering. *Information and Control*. 15, 22-32.

Ruspini, E. H. 1970. Numerical methods for fuzzy clustering. *Information Science*. 2, 319-350.

Salton, G. 1971. *The SMART retrieval system: Experiments in automatic document processing*. Prentice-Hall.

Salton, G. 1989. *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Addison-Wesley, Reading, Mass.

Salton, G. and Lesk, M. E. 1968. Computer evaluation of indexing and text processing. *Journal of the Association for Computational Machinery*. 15, 1, 8-36.

Salton, G. and McGill, M. J. 1983. *Introduction to modern information retrieval*. McGraw-Hill. ISBN 0070544840.

Scott, S. and Matwin, S. 1998. Text classification using WordNet hypernyms. *Usage of WordNet in Natural Language Processing Systems: Proceedings of the Workshop*. 45-51.

Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22, 8, 888-905.

Sinclair, J., ED. 2001. *Collins Cobuild English dictionary for advanced learners*. 3rd ed. Harper Collins.

Sinha, R. and Mihalcea, R. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the IEEE International Conference on Semantic Computing*. (ICSC'07, Irvine, CA). 363-369.

Skabar, A. and Abdalgader, K. 2010. Improving sentence similarity measurement by incorporating sentential word importance. In *Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence*. (AI2010, Adelaide). Advances in Artificial Intelligence. 6464, 466-475.

Skabar, A. and Abdalgader, K. 2011. Clustering sentence-level text using a novel fuzzy relational clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. IEEE computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.205>

Snyder, B. and Palmer, M. 2004. The English all-words task. In *Proceedings of ACL/SIGLEX*. (Senseval-3, Barcelona, Spain). 41–43.

Steyvers, M., Shiffrin R.M. and Nelson, D.L. 2004. Word association spaces for predicting semantic similarity effects in episodic memory. In *A. Healy (Ed.), Cognitive Psychology and its Applications: Festschrift in Honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer*. Washington DC: American Psychological Association.

Stokoe, C. 2005. Differentiating homonymy and polysemy in information retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. (HLT/EMNLP, Vancouver, Canada). 403-410.

Theodoridis, S. and Koutroumbas, K. 2008. *Pattern recognition*. 4th Edn. Academic Press, Burlington MA: USA.

Tsatsaronis, G., Varlamis, I., and Norvag, K. 2010. An experimental study on unsupervised graph-based word sense disambiguation. In *Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics*. (CICLing'10, Iasi, Romania). LNCS 6008, 184–198.

Turney, P. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In De Raedt, Luc and Flach, Peter, Eds. In *Proceedings of the 12th European Conference on Machine Learning* (ECML-2001). 491-502

Van Rijisbergen, C.J. 1975. *Information Retrieval*. Butterworths, London.

Vickrey, D., Biewald, L., Teyssier, M., and Koller, D. 2005. Word-sense disambiguation for machine translation. In *Proceedings of Human Language*

Technology Conference and Conference on Empirical Methods in Natural Language Processing. (HLT/EMNLP, Vancouver, Canada). 771–778.

Vidhya, K. A. and G. Aghila, G. 2010. Text Mining Process, Techniques and Tools: an Overview. *International Journal of Information Technology and Knowledge Management*. 2, 2, 613-622.

Wall, M. E., Andreas R., and Luis M. 2003. Singular value decomposition and principal component analysis. In *A Practical Approach to Microarray Data Analysis*. D.P. Berrar, W. Dubitzky, M. Granzow, eds. 91-109. LANL LA-UR-02-4001.

Wang, D., Li, T., Zhu, S. and Ding, C. 2008. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 307-314.

Wiebe, J. M. 1994. Tracking point of view in narrative. *Computational Linguistics*. 20, 2, 233-287.

Windham, M. P. 1985. Numerical classification of proximity data with assignment measures. *Journal of Classification*. 2,157-172.

Witten, I. H., Moffat, A. and Bell, T. C. 1999. *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann Publishers Inc., 2nd edition. ISBN 1-55860-570-3.

Wu, Z. and Palmer, M. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, New Mexico. 133-138.

Yang, M. S. 1993. A survey of fuzzy clustering. *Mathematical Computer Modelling*.

18, 11, 1-16.

Yang, D. and Powers, D. 2005. Measuring semantic similarity in the taxonomy of WordNet. In *Proceedings of the 28th Australasian Computer Science Conference*. 315- 332.

Yarowsky, D. and Florian, R. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*. 8, 4, 293–310.

Yu, S. X. and Shi, J. 2003. Multiclass spectral clustering. In *Proceedings of International Conference on Computer Vision*. 11-17.

Zha, H. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 113-120.

Zhai, Z., Liu, B., Xu, H. and Jia, P. 2011. Clustering product features for opinion mining. In *Proceedings of WSDM'11*. 347-354.

Zhao, L., Wu, L., and Huang, X.J. 2006. Fudan University at DUC 2006. *Document Understanding Conferences 2006* (DUC '06).

Zhao, M., Wang, J. and Fan, G. 2008. Research on application of improved text cluster algorithm in intelligent QA system. In *Proceedings of the Second International Conference on Genetic and Evolutionary Computing, China, IEEE Computer Society*. 463-466.

Zipf G. K. 1949. *Human behavior and the principle of least effort*. Cambridge, Massachussetts: Addison-Wesley. 1.